

# Self-Monitoring for Sensor Networks

Dezun Dong

Yunhao Liu

Xiangke Liao

## ABSTRACT

Local monitoring is an effective mechanism for the security of wireless sensor networks (WSNs). Existing schemes assume the existence of sufficient number of active nodes to carry out monitoring operations. Such an assumption, however, is often difficult for a large scale sensor network. In this work, we focus on designing an efficient scheme integrated with good self-monitoring capability as well as providing an infrastructure for various security protocols using local monitoring. To the best of our knowledge, we are the first to present the formal study on finding optimized self-monitoring topology for WSNs. We show the problem is NP-complete even under the unit disk graph (UDG) model, and give the upper bound on the approximation ratio. We further propose two distributed polynomial algorithms with provable approximation ratio to address this issue. Through comprehensive simulations, we evaluate the effectiveness of this design.

## Categories and Subject Descriptors

C.2.0 [Computer-Communication Networks]: General – Security and Protection. F.2.2 [Analysis of Algorithms and Problem Complexity]: Nonnumerical Algorithms and Problems – Complexity of proof procedures.

## General Terms

Algorithms, Theory, Security.

## Keywords

Wireless Sensor Network, Security, Self-Monitoring, NP-Complete.

## 1. INTRODUCTION

Wireless sensor networks (WSNs) are emerging as a promis-

Dezun Dong is a PHD student at the School of Computer in National University of Defense Technology, Changsha, Hunan, China, dong@nudt.edu.cn. He is co-supervised by Dr. Yunhao Liu.

Yunhao Liu is with the department of Computer Science in Hong Kong University of Science and Technology, liu@cse.ust.hk. He is also a professor at Xi'an Jiaotong University.

Xiangke Liao is a professor at the School of Computer in National University of Defense Technology, Changsha, Hunan, China, xkliao@nudt.edu.cn.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

MobiHoc'08, May 26-30, 2008, Hong Kong SAR, China.  
Copyright 2008 ACM 978-1-60558-073-9/08/05...\$5.00.

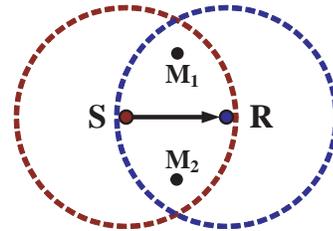


Figure 1. Local monitoring.

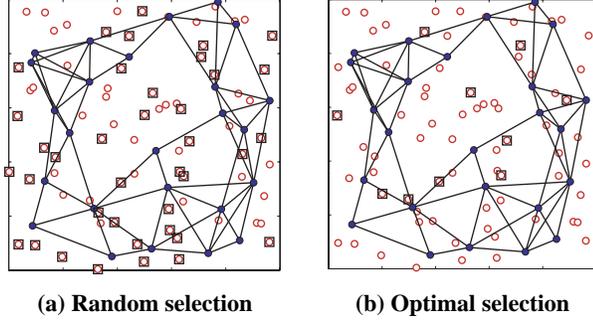
ing platform for many important applications such as military surveillance, homeland security, emergency response, forest fire monitoring, etc. Security is important for mission-critical applications which work in unattended and even hostile environment [1, 2]. One of the most severe security threats in sensor networks is node compromise. Once some nodes are compromised, the attackers can use them to mount a variety of attacks. It is rather challenging to provide effective security mechanisms against compromised nodes in resource-limited WSNs [3].

Based on local monitoring (or watchdog) technique [4-7], many approaches have been proposed to secure sensor networks in face of compromised nodes. The basic idea of local monitoring is illustrated in Fig. 1. The dashed circle denotes the transmission range of a node. Node S,  $M_1$ ,  $M_2$  monitor the link from S to R, since they are able to monitor the traffic that R receives from S and sends to others.

In WSNs, local monitoring is a promising security mechanism as an effective complement for cryptographic mechanisms. Existing local monitoring schemes assume the existence of sufficient nodes to carry out the monitoring function. Such a requirement is often practically difficult when we consider minimizing the number of monitoring nodes selected for a large scale WSN.

For example, as shown in Fig. 2, the dots and lines denote the active sensor nodes and communication links between them. The circles denote the sleeping nodes. We desire that every communication link is monitored by two other nodes except for the end nodes of the link. Note that this requirement cannot be satisfied since there are not enough (two) active nodes neighboring to links. Hence, we have to select additional sleeping nodes and activate them. For independent selection for each link, the nodes are selected randomly among those that are able to monitor a link. Figure 2(a) shows the results of random selection, where 38 additional nodes are selected. As shown in Figure 2(b), only 10 nodes are needed, if we adopt an optimal strategy. The nodes within box are the additional nodes selected to satisfy one communication link monitored by two nodes.

We focus on the fundamental problem of designing a self-monitoring topology, where each communication link can be monitored by nodes within the network. We prove that finding an optimal self-monitoring topology is NP-complete even when we



**Figure 2. Comparison of random and optimal selection.**

model the communication network as a unit disk graph (UDG) with a geometric representation. We provide the approximability results in centralized scenario, and prove the existence of polynomial-time approximation scheme (PTAS) for this problem when restricted in some specific graphs. We develop two efficient and distributed approximation algorithms with provable approximation ratio and time complexity guarantee for large scale sensor networks. Moreover, we conduct extensive simulations to demonstrate the effectiveness of this design.

The rest of the paper is organized as follows. Related work is presented in Section II, and problem definition is given in Section III. Hardness of the problem and approximability results are presented in Section IV. Distributed algorithms are presented in Section V. Performance evaluation is presented in Section VI. We have some discussions in VII, and conclude this work in Section VIII.

## 2. RELATED WORK

WSNs are vulnerable to a wide range of security attacks including wormhole, black hole, spoofed, altered, replayed routing information, selective forwarding, sybil attack, DoS, etc. [2, 3]. There have been many proposals using cryptography to ensure secure communication such as SPINS[1], etc.

Cryptography provides an efficient mechanism to achieve data confidentiality, data integrity, node authentication, and secure routing. Nevertheless, cryptography alone is not sufficient for node compromise attacks and novel misbehaviors in sensor networks [6]. Researchers therefore attempt to use non-cryptographic techniques to solve the problems beyond the capability of cryptographic security [4-7].

Previous studies introduce the concept of watchdog or local monitoring to mitigate attacks in sensor and ad hoc networks [4-9]. The idea of watchdog is first introduced by Marti et al. in ad hoc networks for detecting mischievous nodes [10]. Since then extensive efforts have been made in ad hoc [11-13] and sensor networks [4-7], falling into three categories: secure routing, reputation and trust systems, and intrusion detection.

For secure routing, a lightweight protocol called DICAS is proposed by Khalil et al. [4], which mitigates the control and data traffic attacks in sensor networks with the help of local monitoring. They design a countermeasure for wormhole attacks, called LITEWOP [5], which uses guard nodes to attest the source of each transmission. Neighbor watch [7] is employed by a hop-by-hop resilient packet-forwarding scheme. For reputation and trust-based systems, neighbor watch is used as a component to monitor neighborhoods and collect information to build trust relationships among nodes in the network, such as RFSN[6], CONFIDANT[11],

CORE[13], etc. For intrusion detection systems, local monitoring is used to build decentralized protocols [9, 12].

Local monitoring scheme places new requirement to a sensor connectivity topology. There are two existing topology control schemes: radio power control and wake/sleep schedule, including LEACH [14], SPAN [15], STEM [16], etc. The main goal of those schemes, however, is to achieve energy efficient communication. They aim at extending network lifetime, enhancing network utilization and capacity, minimizing interference, reducing high end-to-end packet delays, and increasing the robustness to node failures. They do not construct and optimize topologies with local monitoring properties.

Recently, Khalil et al. [17] propose a on-demand sleep-wake protocol to shorten the time a node needs to be awake for the purpose of monitoring. They do not, however, consider the optimized selection of monitoring nodes in the network, but focusing on how to schedule nodes to meet the monitoring requirement for given communication links.

It is worth noting that the focus of the self-monitoring mechanism proposed by Hsin et al. [18] is completely different from this work. They pay more attention on the system-level fault diagnosis of the network, especially detecting node failures. They do not deal with malicious behaviors as what are considered in the works [4, 5, 7]. On the other hand, our study emphasizes the optimized node selection for the local monitoring scheme.

## 3. SELF-MONITORING PROBLEM

We consider static (dense) sensor networks in which all the communication links are bidirectional. We define the *communication graph* on the network as a directed graph  $G_c$ , where sensors are represented by vertices. A pair of opposite directed edges  $\langle u, v \rangle$  and  $\langle v, u \rangle$  exist if there is a direct communication channel between node  $u$  and  $v$ . We define the active network created and scheduled by topology control algorithm as *topology graph*, which is a subgraph of communication graph  $G_c$ . In the rest of the paper, we use  $E(G)$  and  $V(G)$  to denote the edge and vertex set of a graph  $G$ , respectively. Node/vertex, as well as link/edge will be used interchangeably. It is important to note that the assumption about the bidirectional communication links is just for convenient presentation of algorithm. Indeed all definitions and theorems in the paper are still applicable when the communication links are directional.

**Definition 3.1** Given a directed edge  $e = \langle v_1, v_2 \rangle \in E(G_c)$ ,  $v \in V(G_c) \setminus \{v_1\}$ , if  $\langle v_1, v \rangle, \langle v_2, v \rangle \in E(G_c)$ , we say  $v$  can *monitor* edge  $e$ . Given an edge set  $E \subseteq E(G_c)$ , a vertex set  $V \subseteq V(G_c)$ ,  $k \in \mathbb{N}$ , if any edge  $e$  in  $E$  can be monitored by at least  $k$  different vertices in  $V$ , we say  $V$  can *k-monitor* the edge set  $E$ .

**Definition 3.2** Given a graph  $G \subseteq G_c$ , and an edge set  $E \subseteq E(G)$ , let  $b \in \mathbb{Z}_{0,+}^E$ , if for any  $e \in E$ , there are just  $b(e)$  different vertices in  $V(G)$  that can monitor  $e$ , we say  $G$  has the *b-self-monitoring* capability about edge set  $E$ . Further, if given  $k \in \mathbb{N}$ , for any  $e \in E$ ,  $b(e) \geq k$ , we say  $G$  has the *k-self-monitoring* capability about edge set  $E$ .

The construction of connectivity topology of WSNs is often optimized for multiple objectives, such as required connectivity and coverage, more specific and application-oriented requirements, and so on. Self-monitoring capability is a new requirement for connectivity topology, for which it is necessary to investigate how

to integrate the self-monitoring sub-objective with previous ones. To have our algorithms generally applicable, we treat self-monitoring as an independent function module, that is, we investigate how to convert a general connectivity topology into a topology with self-monitoring capability at lowest cost. We formalize the transition as follows.

**Definition 3.3** Minimum Self-Monitoring Topology Problem (MSMTP): Given  $G_c, G_0 \subseteq G_c, k \in \mathbb{N}$ , finding a subgraph  $G_l \subseteq G_c$ , such that  $G_0 \subseteq G_l, V(G_l)$  can  $k$ -monitor  $E(G_0)$  and  $|V(G_l)|$  is minimized.

Definition 3.3 is equivalent to adding minimum number of vertices in  $V(G_c) \setminus V(G)$  to satisfy the  $k$ -monitoring requirement, as formally listed in Definition 3.4.

**Definition 3.4** Minimum Patching Monitoring Set Problem (MPMSP): Given  $G_c$  and  $G$ , an edge set  $E_p \subseteq E(G)$  which denotes the edges that cannot be  $k$ -monitored in  $G$ , let  $b \in \mathbb{N}^{E_p}$  be the monitoring number for  $E_p, V_p$  denote  $V(G_c) \setminus V(G)$ , finding a subset of vertices  $V \subseteq V_p$  such that  $V$  can  $b$ -monitor  $E_p$  and  $|V|$  is minimized.

## 4. PROBLEM HARDNESS AND APPROXIMATION

In this section, we first prove that the MPMSP problem is NP-complete even the communication graph is restricted to be UDG with a geometric representation. We discuss the bounds on approximation ratio for MPMSP.

### 4.1 Problem Hardness

The hardness of MPMSP largely depends on the graph model for representing communication topology of a sensor network. Among those models, UDG is probably the simplest one [19]. Hence, this discussion will start from proving MPMSP is NP-complete in UDG with a geometric representation, and then extend it to generalized models.

We use the *proximity model* [20] to define UDG, that is, points in the plane form a UDG with a vertex corresponding to a point and an edge between two vertices exists if and only if the Euclidean distance between the two points is at most a constant bound  $C$ . Before presenting the proof, related definitions and notations are listed as follows.

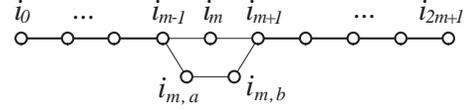
**Definition 4.1** Given  $(G_c, G, E_p, V_p, b)$  of a MPMSP, we define the *monitoring degree* of a vertex  $v$  in  $V_p$  as the number of edges in  $E_p$  that can be  $v$  monitored, denoted by  $\delta(v)$ ; the *monitored degree* of an edge  $e$  in  $E_p$  is defined as the number of vertices in  $V_p$  that can monitor  $e$ , denoted by  $\lambda(e)$ ; and we denote the *maximum monitoring degree* as  $\Delta = \max\{\delta(v), v \in V_p\}$ , and *maximum monitored degree* as  $\Lambda = \max\{\lambda(e), e \in E_p\}$ .

**THEOREM 4.1.** *When  $\Delta \geq 3$ , MPMSP is NP-complete in UDG with a geometric representation.*

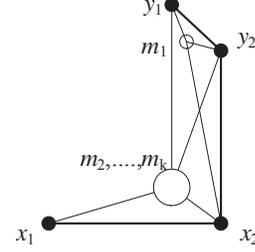
**PROOF.** To show MPMSP is NP-complete, it is sufficient if we prove that when the monitored number  $b$  is specialized to a constant  $k \geq 2$ , the problem is NP-complete. We denote it as  $k$ MPMSP.

To show that  $k$ MPMSP belongs to NP, we need to show whether every edge,  $e$ , in  $E_p$  is  $k$ -monitored by some vertices in  $V \subseteq V_p$ , which can be accomplished in polynomial time.

To prove that  $k$ MPMSP in UDG is NP-hard, we can show that the Vertex Cover problem in planar graph with maximum degree



**Figure 3.** Locally redraw a grid unit.



**Figure 4.** Gadgets  $W$ .

3 is polynomial-time reducible to  $k$ MPMSP, as the former is well-known NP-complete [21].

Hence, we will present a polynomial-time transformation that takes an arbitrary planar graph  $G_a$  of maximum degree 3 and constructs a  $k$ MPMSP  $(G_c, G, E_p, V_p, k)$  in UDG with a geometric representation and  $\Delta \geq 3$ . Moreover, knowing minimum patching monitoring set of  $(G_c, G, E_p, V_p, k)$ , we are able to compute minimum vertex cover of  $G_a$  in polynomial time.

The key issue for the construction of UDG  $G_c$  is the selection of points set and the distance bound  $C$ , which can be carried out in two steps.

**Step 1.** We first draw  $G_a$  in a plane by constructing a planar orthogonal grid embedding [22] for it. The embedding maps (1) vertices to distinct grid points, and (2) edges to nonintersecting grid paths. All vertices and bends are located on integer grid points. Such a construction of a planar orthogonal grid embedding does exist [23] and can be finished in polynomial time  $O(|V(G_a)|)$  [22]. Second, we enlarge the scale such that the unit length of grid is  $2m+1, m \in \mathbb{N}$ , and  $m \geq 4$ . Let  $|e|$  denote the length of embedding image of  $e \in E(G_a)$ , for any  $e_i \in E(G_a)$ ,  $|e_i|$  is a multiple of  $2m+1$ . Here we use one notation to denote both vertex/edge and its drawing. Third, for any  $e_i \in E(G_a)$ , if  $|e_i|$  is even, we select a segment of unit grid in  $e_i$  and redraw it locally such that  $|e_i|$  increments by 1 and becomes an odd, as illustrated in Fig. 3. The line segment of length  $2m+1$  from  $i_0$  to  $i_{2m+1}$  denote a grid unit. Delete the segment  $(i_{m-1}, i_{m+1})$  of length 2, reconnect  $i_{m-1}, i_{m+1}$  by three segments of length 1,  $(i_{m-1}, i_{m,a}), (i_{m,a}, i_{m,b}), (i_{m,b}, i_{m+1})$ . After redrawing, the embedding of  $G_a$  is no more an orthogonal grid embedding. We then insert all the vertices of  $G_a$  to  $G$  and  $G_c$ .

**Step 2.** For an edge  $e_i \in E(G_a)$ , we place a number  $L = |e_i| - 1$  of vertices on  $e_i$ , by making the interspaces of two adjacent vertices be equal to 1, and denote them as  $v_{i,l}, l \in [1, L]$ . We add all these vertices into  $G$  and  $G_c$ . Since  $|e_i|$  is odd,  $|e_i| - 1$  is even. Let  $v_{i,0}, v_{i,L+1}$  denote the endpoints of  $e_i$ . Now, for each vertex  $v_{i,l}$  in  $G, l \in [1, L]$ , we add a vertex  $u_{i,l}$  such that  $d(v_{i,l}, u_{i,l}) < R$ , where  $R$  is a constant. We discuss the value of  $R$  later. We add all these vertices  $u_{i,l}$  to  $G_c$ . If we set  $R=0.1, C=1.1$ , it is straightforward to verify that the following inequations can be satisfied:

- 1)  $\forall e_i \in E(G_a), l \in [1, L],$   
 $d(u_{i,l}, v_{i,l-1}), d(u_{i,l}, v_{i,l}), d(u_{i,l}, v_{i,l+1}) < C$   
 $d(v_{i,l}, v_{i,l-1}), d(v_{i,l}, v_{i,l+1}) < C$
- 2)  $\forall e_i \in E(G_a), l \in [1, L], \forall v \in V(G) \setminus \{v_{i,l-1}, v_{i,l}, v_{i,l+1}, u_{i,l}\}$   
 $d(u_{i,l}, v) > C$

3)  $\forall e_i \in E(G_a), l \in [1, L], \forall v \in V(G) \setminus \{v_{i,l-1}, v_i, v_{i,l+1}\}$   
 $d(v_{i,l}, v) > C$

Step 3. For each edge in  $G$ , say  $(v_{i,l}, v_{i,l+1})$ , we place vertices  $m_{i,l,1}, \dots, m_{i,l,k}, y_{i,l,1}, y_{i,l,2}$  nearby the edge, and adjust the location of those vertices properly such that  $\text{Graph}[v_{i,l}, v_{i,l+1}, y_{i,l,1}, y_{i,l,2}, m_{i,l,1}, \dots, m_{i,l,k}]$  (or  $\text{Graph}[v_{i,l+1}, v_{i,l}, y_{i,l,1}, y_{i,l,2}, m_{i,l,1}, \dots, m_{i,l,k}]$ ) with neglecting edges  $(m_{i,l,r}, m_{i,l,s}), r, s \in [1, k]$ , is isomorphic to gadgets  $W[x_1, x_2, y_1, y_2, m_1, \dots, m_k]$  as shown in Fig. 4. The lines denote the connection relationship. Note that each  $m_i, i \in [2, k]$ , is connected to  $x_1, x_2, y_1, y_2$ , but not all the lines are drawn for concision. We restrict that only one of the two vertices,  $v_{i,l}$  and  $v_{i,l+1}$ , to be qualified for mapping to  $x_2$  in  $W$ , and we denote it as  $x_{i,l,2}$  and the other as  $x_{i,l,1}$ . We insert all the vertices into  $G_c$ , while only insert  $y_{i,l,1}, y_{i,l,2}$  into  $G$ . Also, we constrain that the edge  $(y_{i,l,1}, y_{i,l,2})$  is only able to form a  $K_3$  graph with vertices  $m_{i,l,1}, \dots, m_{i,l,k}$  among all vertices in  $G_c$ , and  $m_{i,l,1}, \dots, m_{i,l,k}$  is only able to form a  $K_3$  with edges  $(x_{i,l,1}, x_{i,l,2}), (x_{i,l,2}, y_{i,l,1}), (y_{i,l,1}, y_{i,l,2})$  among all edges in  $G$ . Clearly, all the constraints can be achieved by properly adjusting the positions of new added vertices when setting the parameter values  $R$  and  $C$  as in step 2.

Till now we have finished the transformations in polynomial time. Moreover, our constructed graph  $G_c$  and  $G$  are both connected UDG with the geometric representation, and  $G$  is a subgraph of  $G_c$ . We can check that  $G$  has the 0-self-monitoring capability about any edge in  $E(G)$ . To make  $G$  have the  $k$ -self-monitoring capability about  $E_p = E(G)$ , we can set  $V_p = V(G_c) \setminus V(G)$ . Thus, we obtain the  $k$ MPMSP  $(G_c, G, E_p, V_p, k)$ . It is trivial to verify that maximum monitoring degree  $\Delta$  is 3 in our constructed problem instance.

Finally, it is easy to verify that  $G_a$  has a vertex cover set of size  $N$  if and only if there is a patching monitoring set of  $M = N + \sum_{e_i \in E(G_a)} (k+1/2)(|e_i| - 1)$  size for the  $k$ MPMSP  $(G_c, G, E_p, V_p, k)$ . Thus, the NP-completeness of  $k$ MPMSP is proved. ■

The UDG model is idealistic. Some researchers have proposed other relaxed models for sensor network, such as, quasi unit disk graph model, bounded independence graph, unit ball graph, UDG with hop interference, and general graph [24]. We call those graphs *extended UDG* since UDG is a subgraph of them.

**COROLLARY 4.2.** *When  $\Delta \geq 3$ , MPMSP is NP-complete in extended UDG.*

**THEOREM 4.3.** *When  $\Delta \leq 2$ , MPMSP is polynomial-time solvable in general graph.*

**PROOF.** When  $\Delta \leq 2$ , MPMSP can be formalized as the *simple b-edge covers* on multigraph, which is polynomial-time solvable [25]. Given MPMSP  $(G_c, G, E_p, V_p, b)$ , the procedures of constructing multigraph  $G_m$  are as follows. An edge  $e \in E_p$  corresponds to a vertex  $v_e \in V(G_m)$ . If a vertex  $v \in V_p$  can monitor only one edge  $e \in E_p$ , we add a distinct loop to  $v_e \in V(G_m)$ . If a vertex  $y \in V_p$ , can monitor both edge  $f, g \in E_p$ , we also add a distinct edge  $(v_f, v_g)$  to  $G_m$ . ■

## 4.2 Approximability Results

**THEOREM 4.4.** *There exists  $\rho$ -approximation algorithm for MPMSP in general graph, where  $\rho = \min(H(\Delta), \Lambda)$ .*

**PROOF.** When we disposal the MPMSP in general graph model, clearly, the MPMSP can be formalized as the set multi-cover problem. Hence we can acquire the approximation ratio from set multi-cover [26].  $H(n) = \sum_{i=1}^n 1/i$  is the  $n$ th harmonic number. ■

**THEOREM 4.5.** *There exists a polynomial-time approximation scheme for MPMSP in UDG with a geometric representation.*

**PROOF.** Given  $(G_c, G, E_p, V_p, b)$ , when  $G_c$  is a UDG with geometric representation, all vertices that can monitor the same edge lie in the lune region which is the intersection of two disks with their centers the endpoints of the monitored edge. As a result, every edge  $e$  in  $E_p$  is corresponding to (one-to-one) a lune  $l_e$  in the plane, and the MPMSP can be considered as selecting minimum size of vertices set from  $V_p$  to hit the lunes such that each lune, say  $l_e$ , is hit  $b(e)$  times.

We design a *shifting strategy* [27] to approximate MPMSP. Let  $R$  denote the least rectangular region in which graph  $G_c$  can be drawn. For a positive integer  $m > 0$ , and even integers  $i, j$ , where  $2 \leq i, j \leq m$ , we partition the region  $R$  into squares by horizontal lines at  $l \equiv i \pmod m$  and vertical lines at  $t \equiv j \pmod m$ . Let  $S_{i,j}$  denote the set of squares for a fixed pair  $i, j$ . Let  $S$  denote the union of all the  $S_{i,j}$ . A lune is said to be belonging to a square if and only if its geometric centre lies in the square. For square  $s \in S_{i,j}$ , let  $L(s)$  denote the set of lunes belonging to the square  $s$ .

We assume the node density of the network has an upper bound, so the number of nodes in each  $m \times m$  square is  $O(m^2)$ . Thus, the optimal solution in each square can be found in polynomial time using complete enumeration for a given constant  $m$ . The union of those sets gives a candidate hitting set for fixed  $i, j$ . By changing the parameter  $i, j$ , we have the minimum set.

Now we analyze the approximation ratio. Let  $H_o$  be an optimal hitting set, and let  $H$  be the set obtained by the shifting strategy. For fixed  $i, j$ , let  $H_o(i, *), H_o(*, j), H_o(i, j)$  respectively be the vertices set in  $H_o$  and lie in lunes intersecting horizontal active lines, vertical active lines, and both horizontal and vertical active lines. Let  $H_o(s)$  be vertices in  $H_o$  and in  $L(s)$ ,  $OPT(s)$  be the optimum hitting set for lunes  $L(s)$ . We have

$$|H| \leq \sum_{s \in S_{i,j}} OPT(s) \leq \sum_{s \in S_{i,j}} |OPT(s)| \leq \sum_{s \in S_{i,j}} |H_o(s)|$$

Since vertices in lunes that hit an active line can be used in at most four squares, we have

$$\sum_{s \in S_{i,j}} |H_o(s)| \leq 3 |H_o(i, j)| + |H_o|$$

Note that we set the shifting step be two units, so that all lunes that hit one horizontal(or vertical) active line do not intersect with lunes that hit another horizontal(or vertical) active line. Consequently, we obtain the following inequations,

$$\sum_{2 \leq i \leq m} |H_o(i, *)| \leq |H_o|, \sum_{2 \leq j \leq m} |H_o(*, j)| \leq |H_o|$$

There exist some choices of  $(i, j)$ , such that

$$|H_o(i, *)| \leq 2 |H_o| / m, |H_o(*, j)| \leq 2 |H_o| / m$$

For the choice of  $(i, j)$ , we have

$$|H_o(i, j)| = |H_o(i, *) \cup H_o(*, j)| \leq |H_o(i, *)| + |H_o(*, j)| \leq 4 |H_o| / m$$

And then,

$$|H| \leq (1 + 12/m) |H_o|$$

Hence, given  $\varepsilon > 0$ , let  $m > 0$  be the smallest even integer such that  $(12/m) \leq \varepsilon$ , the solution  $H$  has  $1 + \varepsilon$  approximation ratio. ■

## 5. DISTRIBUTED ALGORITHM

As a large-scale sensor network typically work in a distributed and ad hoc manner, in this section we present two localized algorithms for the MPMSP problem, called *local maximal element* (LME) and *locally dual-feasible* (LDF). None of them uses location information, and they are independent of communication models.

For a general communication network  $G \subseteq G_c$ , we assume that the self-monitoring requirements for each link is known for all the nodes, or is specified by the upper layer protocols using the self-monitoring as an underlying function.

Each node in  $G$  exchanges the neighbor list with one-hop neighbors. Thus, a node  $P$  can determine which links can be monitored by  $P$  and which nodes can monitor the links adjacent (connecting) to  $P$ . The links that are lower to the self-monitoring requirement form the  $E_p$ , and monitoring number of each link,  $b$ , is determined accordingly. All nodes in  $V(G_c) \setminus V(G)$  form the set  $V_p$ .

Given  $G_c$  and  $G$ , a MPMSp  $(G_c, G, E, V, b)$  is denoted as MPMSp  $(E, V, b)$ , in which two vertices in  $V$  are said to be *adjacent* if they are monitoring the same edge in  $E$ . The *adjacent vertex set* of  $v$ , denoted as  $A(v)$ , is the set of all the vertices adjacent to  $v$  in  $V$ . The *monitoring degree* of  $v$ , denoted as  $\delta(v)$ , is the number of edges in  $E$  that can be monitored by  $v$ .

### 5.1 Local Maximal Element Algorithm

For a MPMSp  $(E, V, b)$ , LME deals with the problem from the candidate vertices in  $V$ . A candidate vertex is chosen only if it is optimal within its adjacent vertex set. The priority of a vertex depends on its monitoring degree. It means that a vertex has higher priority if its monitoring degree is high. We break the tie by selecting the vertex with the largest ID among these vertices.

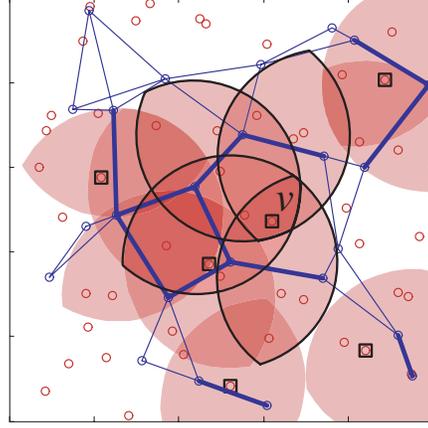
LME is carried out in a parallel manner. In each round, all the locally optimal vertices are selected. The monitored degree of every edge and candidate monitoring vertex set are updated accordingly. LME is run iteratively until the solution is found or the candidate monitoring vertex set  $V$  is empty. Note that a loosely synchronous clock among local vertices is adequate for LME, instead of a global synchronization.

As illustrated in Fig. 5, where the communication graph are modeled as UDG, the vertex having higher monitoring degree means it falls into the overlap field of more lunes. The circles denote the sensor nodes, and the lines denote the links in the connectivity topology. The bold lines denote the edges that do not meet the 1-self-monitoring requirement. The node  $v$  that falls in the overlap of 4 lunes with borderline is first selected into the patching monitoring set, for it can monitor more edges than its adjacent nodes. The squares denote a solution found by LME for 1-self-monitoring.

**THEOREM 5.1.** *The LME algorithm computes a  $H(\Delta)$  approximation for MPMSp in general graph within  $O(\Delta)$  rounds w.h.p.*

We later will present a strict proof to show LME is a faithful implementation of the sequential greedy algorithm. The conclusion somehow counters to the intuition, since sequence greedy selects the globally optimal node in each step while LME only selects locally maximal nodes in each round.

Given an instance  $(E, V, b)$ , each element in  $V$  has a priority value. Now we define a strict partial order relationship “ $\succ$ ” on  $V$ . Relationship “ $\succ$ ” is irreflexive and transitive, defined by: for  $u, v \in V$ , if the priority of  $u$  is higher than that of  $v$  and the selection of one will decrease the priority of the other directly, then  $u \succ v$ . If  $u \succ v, v \succ w$  for  $u, v, w \in V$ , then  $u \succ w$ . Thus, set  $V$  with the strict order “ $\succ$ ” forms a poset. Apparently, such a poset is not necessarily total. Intuitively, the strict order “ $\succ$ ” symbolizes the direct or indirect dependence relationship among candidate nodes in  $V$ . We further de



**Figure 5. Local maximal element algorithm.**

fine a term *top-antichain* as the set of all maximal elements<sup>1</sup> in  $V$ .

Given  $(E, V, b)$ , we use  $(E(v), V(v), b(v))$  to denote the reduced problem instance after adding the vertex  $v \in V$  into the patching monitoring set.

**LEMMA 5.2.** *Given  $(E, V, b)$ , let  $C_1 = [c_1, c_2, \dots, c_{k-1}, c_k, c_{k+1}, \dots, c_N]$  be a greedy solution sequence for  $(E, V, b)$ ,  $c_k$  be a maximal element in  $V$ , then  $C_2 = [c_1, c_2, \dots, c_{k-1}, c_{k+1}, \dots, c_N]$  is a greedy solution sequence for  $(E(c_k), V(c_k), b(c_k))$ .*

**PROOF.** We use  $(E_{l,i}, V_{l,i}, b_{l,i})$  to denote the reduced problem of  $(E, V, b)$  after selecting the vertices of  $\{c_l \in C_1, l \in [1, i]\}$  and adding them into the patching monitoring set, and use  $P_{l,i}(v)$  to denote the priority of node  $v$  in  $(E_{l,i}, V_{l,i}, b_{l,i})$ . We use  $(E_{2,i}, V_{2,i}, b_{2,i})$  to denote the reduced problem of  $(E(c_k), V(c_k), b(c_k))$  after selecting the vertices of  $\{c_l \in C_2, l \in [1, i], l \neq k\}$  and adding them into the patching monitoring set, and use  $P_{2,i}(v)$  to denote the priority of node  $v$  in  $(E_{2,i}, V_{2,i}, b_{2,i})$ .

Since  $C_1$  is a solution for  $(E, V, b)$ , it is clear  $C_2$  is a solution sequence for  $(E(c_k), V(c_k), b(c_k))$ . Thus we only need to show that  $C_2$  is a greedy sequence, that is,  $P_{2,i}(c_i) > P_{2,i}(c_j)$  for  $c_i, c_j \in C_2$  and  $i < j$ . Since  $c_k$  is a maximal element in  $V$ , the selection of  $c_k$  will not change the priority of  $c_i$  for  $i \in [1, k-1]$ . Otherwise, let  $c_t, t \in [1, k-1]$  be the first element in  $C_1$  which can change the priority of  $c_k$ , we have  $P_{1,i}(c_i) \geq P_{1,i}(c_t) > P_{1,i}(c_k) = P_{1,i}(c_k)$ ,  $c_t \succ c_k$  in  $V$ , so that  $c_k$  will not be a maximal element in  $V$ , contradiction. Hence,  $P_{2,i}(c_i) = P_{1,i}(c_i)$  for  $i \in [1, k-1]$ . Moreover, we have  $P_{2,i}(c_i) = P_{1,i}(c_i)$  for  $i \in [k, N]$ . Note that  $C_1$  is a greedy sequence for  $(E, V, b)$ , so  $P_{2,i}(c_i) = P_{1,i}(c_i) > P_{1,j}(c_j) = P_{1,j}(c_j)$  for  $i, j \in [1, k-1] \cup [k, N]$  and  $i < j$ . ■

**LEMMA 5.3.** *Let  $C_1 = [c_1, c_2, \dots, c_{k-1}, c_k, c_{k+1}, \dots, c_N]$  be a greedy solution sequence for  $(E, V, b)$ ,  $T \subseteq C_1$  is a top-antichain in  $V$ , then  $C_2 = C_1 \setminus T$  is a greedy solution sequence for  $(E(T), V(T), b(T))$ .*

**PROOF:** Through exercising Lemma 5.2 recursively. ■

**THEOREM 5.4.** *LME runs in  $O(\Delta)$  rounds w.h.p., and the solution of LME is equivalent to that of centralized greedy algorithm.*

**PROOF.** Given a MPMSp  $(E, V, b)$ , let the solution of centralized greedy be  $C$ , the solution of LME be  $D$ , and LME runs in  $N$  rounds. Let  $R_i$  denote the selection of LME in  $i$ th round, we have  $D = \bigcup_{i=1}^N R_i$ . Further, let  $(E_i, V_i, b_i)$  denote the reduced problem after selecting

<sup>1</sup>  $u \in V$  is a maximal element means that there does not exist  $v \in V$  such that  $v \succ u$ .

nodes set  $\cup_{i=1}^{i-1} R_i$  and  $C_i$  be greedy solution sequence for  $(E_i, V_i, b_i)$ . Apparently,  $(E_i, V_i, b_i)$  is same to  $(E, V, b)$ .

Firstly, we show that  $R_i$  is a top-antichain in  $V_i$  by proving that (1)  $\forall r \in R_i$ ,  $r$  is a maximal element of  $V_i$ , and (2)  $\forall s \in V_i$  if  $s$  is a maximal element of  $V_i$ ,  $s \in R_i$ . Since all nodes that can affect  $r$  must be in  $A(r) \cap V_i$ , and the priority of  $r$  is the highest in  $A(r) \cap V_i$ ,  $r$  is a maximal element in poset  $(V_i, \succ)$ . Also,  $\forall s \in V_i$  if  $s$  is a maximal element in  $V_i$ , which means that the priority of  $s$  is the highest in  $A(s) \cap V_i$ , so  $s$  will surely be selected by LME algorithm in  $i$ th round. We have  $s \in R_i$ .

Secondly, we show that  $R_i \subseteq C_i$  and  $C_{i+1} = C_i \setminus R_i$ ,  $\forall r \in R_i$ , for an edge which can be monitored by  $r$ ,  $r$  will hold the highest priority among all the vertices that can monitor the edge, so centralized greedy will definitely select  $r$  to monitor that edge, thus  $r \in C_i$ ,  $R_i \subseteq C_i$ . Likewise,  $R_i$  is a top-antichain in  $V_i$ , therefore, from Lemma 5.3 we know  $C_i \setminus R_i$  is a greedy solution sequence for  $E_i(R_i) = E_{i+1}$ ,  $C_{i+1} = C_i \setminus R_i$ .

Thirdly, we show  $D=C$ . Since LME ends in  $N$  rounds, when LME runs in the  $N$ th round for instance  $(E_N, V_N, b_N)$ , it is clear that  $D_N = R_N = C_N$ . Hence,  $C=C_1 = R_1 \cup C_2 = R_1 \cup R_2 \cup C_3 = \dots = R_1 \cup \dots \cup R_N$ ,  $1 \cup C_N = R_1 \cup \dots \cup R_{N-1} \cup D_N = R_1 \cup \dots \cup R_{N-1} \cup R_N = D$ .

Finally, we show that  $N$  is at most  $O(\Delta)$ . Since  $R_i$  is a top-antichain in  $V_i$ , we have  $\forall r_{i+1} \in R_{i+1}$ ,  $\exists r_i \in R_i$  such that  $r_{i+1} \succ r_i$  in  $V_i$ . We say  $r_i$  is the tight-upper vertex of  $r_{i+1}$ . Let  $\delta_i(r)$  denote the monitoring degree of  $r$  in  $i$ th round, then  $\delta_{i+1}(r_{i+1}) \leq \delta_i(r_{i+1}) \leq \delta_i(r_i)$ . If we first choose an arbitrary vertex  $r_N \in R_N$ , then select the tight-upper vertices of  $r_N$ , denoted as  $r_{N-1}$ . Similarly, the tight-upper vertices  $r_{N-2}, \dots, r_1$  are obtained recursively. Thus, we get the following inequality:  $1 \leq \delta_N(r_N) \leq \dots \leq \delta_i(r_i) \leq \dots \leq \delta_1(r_1) \leq \Delta$ . Let the length of maximum equivalence sequence among  $\delta_i(r_i)$  be  $L$ , we have  $N \leq \Delta L$ . If we assume that the nodes are deployed randomly, such that node IDs are distributed randomly, clearly the expectation of  $L$  is  $O(1)$ . ■

It is not difficult to see that Theorem 5.1 can be obtained from Theorem 5.4 since the approximation ratio of sequential greedy algorithm is  $H(\Delta)$  as shown in Theorem 4.4. Note that the bound  $H(\Delta)$  is asymptotically reachable, and such problem instances can be constructed by properly modifying the proof of Theorem 5 [28].

Liang *et. al* have partially observed the similar results we prove here. Their proof [29], however, has some defects, and is not strict but based on intuitive arguments. Indeed, their major statement is equivalent to our work of  $R_i \subseteq C_i$ , as shown in the second step of the proof for Theorem 5.4, while our method of proof is more general.

## 5.2 Local Dual-Feasible Algorithm

The LDF algorithm is different from LME in that the LME considers the solution from the monitoring nodes, while the LDF algorithm deals with the problem from the view of the monitored edges.

An important technique in LDF is the construction of the *edge-dependent* graph. An edge-dependent graph  $G_e$  is derived from the problem  $(E, V, b)$ , and is constructed as follows. A vertex in  $V(G_e)$  corresponds to an edge in  $E$ , and an edge between any two vertices in  $V(G_e)$  exists if the two edges in  $E$ , which corresponding to the two vertices, can be monitored by a common vertex in  $V$ .

LDF is also carried out in rounds, and each round consists of four steps. Let  $(E', V', b')$  denote the reduced problem instance for the current round. First, the derived edge-dependent graph of  $(E', V', b')$  is constructed. Second, calculate a maximal independent set (MIS)

of the derived edge-dependent graph in a distributed manner (There is a substantial research literature on MIS construction). Note that since the vertices in derived edge-dependent graph are one-to-one with an edge in  $E'$ , a subset of  $E'$ , which corresponds to the calculated MIS, can be obtained accordingly. We call such a subset *maximal independent edge set* (MIES) of  $E'$ . Third, for each edge, say  $e$ , in MIES constructed in step two, an expected number of vertices are selected among all the vertices in  $V'$  that are able to monitor the edge. The expected number is equal to the current monitoring number of the edge  $e$ ,  $b'(e)$ . Fourth, each edge in  $E'$  updates its monitored degree according to the newly selected monitoring vertices in step three. At the same time,  $E', V'$  and  $b'$  are all updated accordingly. The LDF algorithm terminates when the edge set  $E'$  is empty.

Let  $T_M$  denote the maximal time of constructing an MIS in each round. We discuss the approximation ratio and time complexity of LDF algorithm as follows.

**THEOREM 5.5.** *The LDF algorithm provides a  $\Lambda$ -approximation for MPMSP in  $O((\Lambda+1)T_M)$  times.*

**PROOF.** The approximation ratio can be acquired from the similar proof for the set multi-cover [30]. We will mainly consider the time complexity. We only need to determine for how many round the edge-dependent graph will be empty. Let  $v$  be a vertex in the derived edge-dependent graph and its degree be  $d(v)$ . We claim vertex  $v$  must be selected into one MIS within  $d(v)+1$  round. If vertex  $v$  is not selected into MIS in a round, there must be at least a vertex neighboring to  $v$  in edge-dependent graph which will be selected into the MIS. Hence, even assuming that  $v$  is the last one being selected among all its neighbors, all neighbors of  $v$  will be selected within at most  $d(v)$  rounds, and finally  $v$  will be selected within at most  $d(v)+1$  rounds. Obviously the maximal degree of the derived edge-dependent graph is  $\Lambda$ . Hence, LDF will end within at most  $\Lambda+1$  rounds. ■

It is worth noticing the difference between LME and LDF on approximation ratio and running time. When changing the problem instances, neither one will be always better than the other one about the approximation ratio and running time. When the maximum monitored degree  $\Lambda$  is small (e.g.  $\Lambda=2$  or 3) and the maximum monitoring degree  $\Delta$  is relatively large, the LDF with the approximation ratio  $\Lambda$  has an advantage over LME with the logarithmic ratio  $H(\Delta)$ . Whereas  $H(\Delta)$  is often preferred rather than  $\Lambda$  for other instances. To some extent, the two algorithms complement each other in the sense of approximation ratio and running time. For example, it is not difficult to check that LDF works very well for the worst case of LME, and vice versa.

## 6. PERFORMANCE EVALUATION

We are not able to simulate the infinite large network, so the objective of our simulations and evaluations is to provide some intuitive results under a feasible range.

Throughout the simulations, we distribute the nodes in a square field, and assume each node has the same transmission radius. We generate the node locations within the field according to two-dimensional uniform random distribution. Initially, for a given topology density, we create the general connectivity topology by randomly selecting some nodes from all the nodes. Topology density (TD) is defined as the average number of one node's neighbors in the active connectivity topology. The worker ratio (WR) is defined as the ratio of the total number of nodes to the

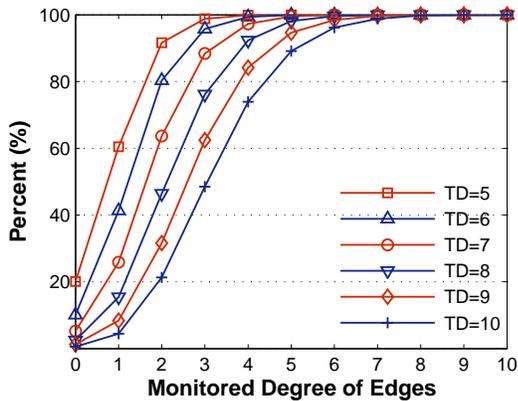


Figure 6. Monitored degree of edges in different topology density.

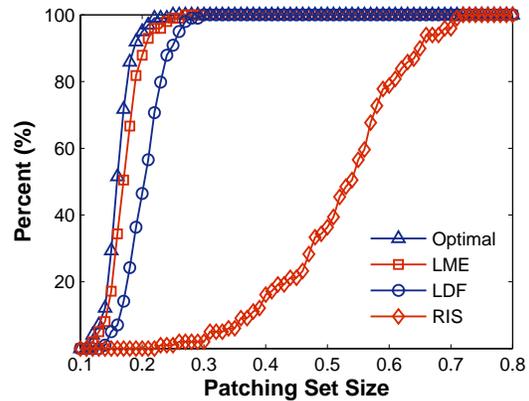


Figure 7. Patching set size of different algorithm.

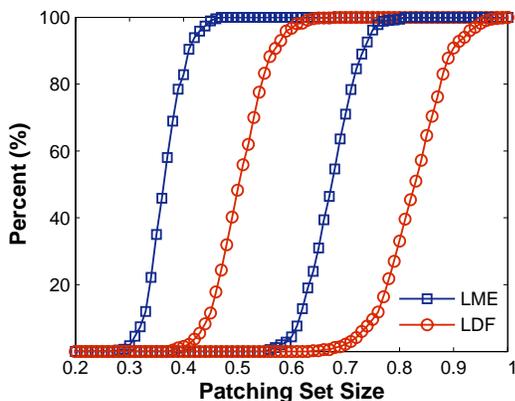


Figure 8. A detailed comparison of LME and LDF.

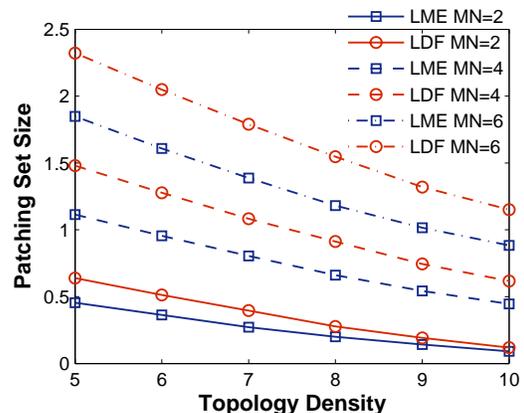


Figure 9. Patching set size against topology density.

number of nodes in connectivity topology. Another important parameter is the monitoring number (MN), which is set a positive integer in the simulations. We run 100 experiments independently to obtain the simulation results as follows. By default, we set the WR to be 6.

## 6.1 Self-Monitoring Capability of Random Topology

We illustrate the self-monitoring capability through the relationship between monitored degree of edge and TD. We set the working topology size of 400 nodes and increase TD from 5 to 10. Figure 6 shows the Cumulative Distributed Function (CDF) of monitored degree of an edge. We can see that as TD increases, the corresponding topology provides more monitored degree for edges. For example, TD of 5 yields 80 percents of edges with monitored degree of 1.5 (and below), and TD of 9 yields 80 percents of edges with monitored degree of 4.0 (and below), which means for a given self-monitoring degree requirement, the higher topology density, the lower proportion of edges needs the additional patching nodes, and vice versa. This is consistent with our intuitions.

## 6.2 Comparison of LME and LDF

We compare our LME and LDF algorithms with the optimal solution and a random algorithm. The optimal solution is obtained

by Matlab Binary Integer Programming tools. The random algorithm used for comparison purpose is called Random Independent Selection (RIS) algorithm. In RIS, for a given edge, the required number of nodes is selected randomly among all the surrounding nodes that can monitor the edge, and all of the edges that need to be monitored make their decisions independently and simultaneously, and add all the selected nodes into the patching monitoring set. Define patching set size as the ratio of the number of nodes in patching monitoring set size to the number of nodes in the initial connectivity topology.

In Fig. 7, we simulate the CDF of set size for four algorithms when TD=8, WN=100, MN=4, WR =5. The results suggest that our LME and LDF are very close to the optimal solution in terms of patching set size. Furthermore, from Fig. 7, we can see the Optimal, LME and LDF demonstrate good threshold phenomenon. We can see that the average performance is better than the theoretical worst case greatly. At the same time, obviously, RIS is the worst and hence it is worthy and necessary to use our algorithm to select the patching monitoring nodes. We summarize the average set size and the ratio against the optimal solution in Table 1. LME and LDF yield very close set size to the optimal size, but RIS's size is more than 3 times compared to the optimal one.

Figure 8 further illustrates the comparison of LME and LDF when changing the TD and MN. The left two lines denote the

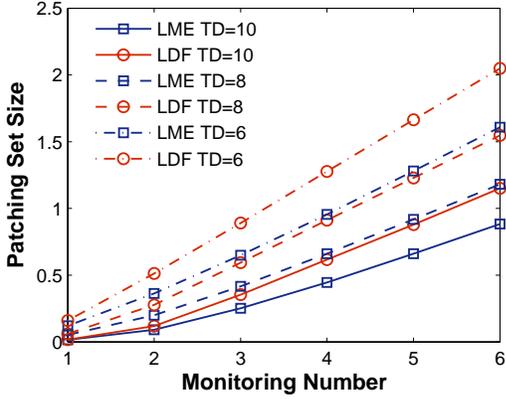


Figure 10. Patching set size against monitoring number.

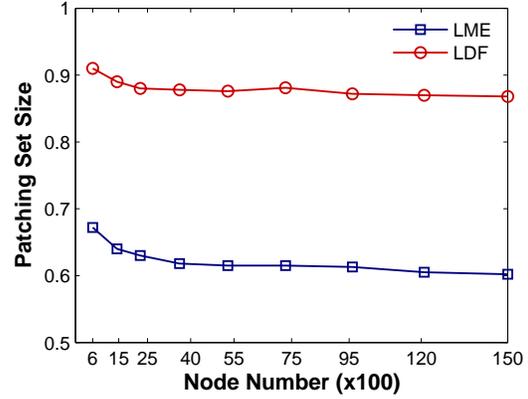


Figure 11. Patching set size against the network size.

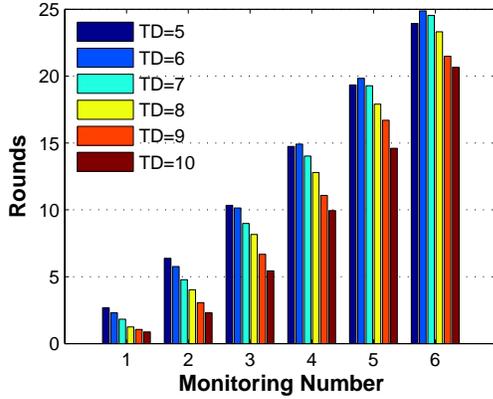


Figure 12. Rounds of LME against monitoring number.

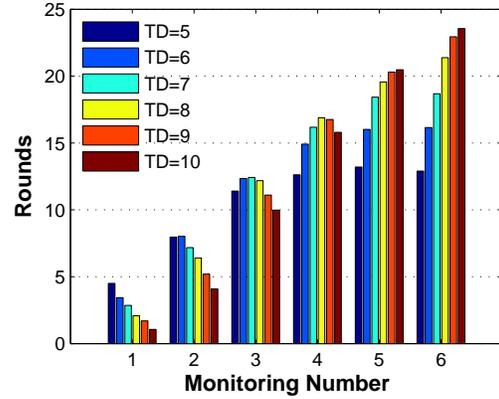


Figure 13. Rounds of LDF against monitoring number.

results when  $TD=6$ ,  $MN=2$ ,  $WR=5$ , the right two lines denote the results when  $TD=8$ ,  $MN=4$ ,  $WR=5$ . In both cases, LME generates better performance than LDF. According to our results, in most of circumstance, LME is superior to LDF, but not always. We change the number of deployed nodes to check the scalability of the algorithms. Figure 11 shows the patching set size patching set size against the number of nodes from 600 to 15000 with  $TD=8$ ,  $MN=4$  and  $WR=6$ . We can see that the results of our two algorithms keep stable when the network scale becomes large.

Table 1. Patching Set Size and Ratio

	Optimal	LME	LDF	RIS
Patching Set size	0.6760	0.7196	0.8508	2.1110
Ratio	1.0000	1.0645	1.2586	3.1229

### 6.3 Impact of Topology Density and Monitoring Number

Since the self-monitoring capability of a topology change with TD as shown in Fig. 6, and the patching set size depends on the TD, we present the simulation results on the patching set size against TD. Figure 9 shows the results where TD changes from 5

to 8 with  $MN=2,4,6$ ,  $WR=6$ . From Fig. 9, we can see the patching set sizes of LME and LDF decrease with the increase of TD, which matches the intuition. Since higher TD means fewer edges to be monitored and the same number of nodes can monitor more edges. Further, we can observe that with the increase of TD, the gap of LME and LDF is gradually diminished, because when TD grows, the edges to be monitored decrease, and more isolated edges or small connected branches emerge, so the solution space that can be exploited for optimization shrinks accordingly. Intuitively, for these isolated edges, the results of two algorithms will be closed to each other.

Similarly, we present the patching set size against monitoring number. In Fig. 10, we can see that the output of LME and LDF almost increase linearly with monitoring number when  $MN \geq 3$ , but grow mildly from 1 to 2, especially for relatively higher  $TD=8$  and 10. This is due to that when  $TD=8$  and 10, only a small part of edges, about 10% and less, have the monitored degree less 2, while most of the edges, about 80% and more, have the monitored degree less than and equal 4, as shown in Fig. 6.

Further, from Fig. 10, we can see that to achieve 2-self-monitoring topology, we only need to add a small size of patching nodes, changing with topology density. It means that self-topology can be accommodated in sensor network with the modest number of extra nodes, especially when TD is relatively high.

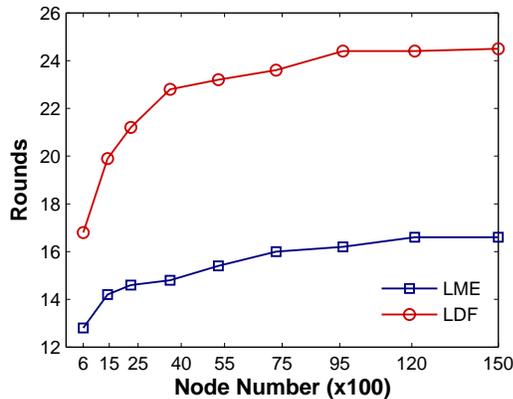


Figure 14. Rounds of LME and LDF against network size.

## 6.4 Time Complexity

We simulate the LME and LDF in the synchronous running manner to investigate their time complexity. For the algorithm of distributed construction of MIS in LDF, we use a simply greedy method. We assign a distinct ID to each to be monitored edge, and the edge with the maximal ID first adds itself into the MIS.

As shown in Fig. 12 and 13, we can see that the running rounds of LME and LDF increase with the monitor number. This is because there are more edges need to be monitored when the monitor number increases. The performance of LME and LDF, however, is quite different from each other, especially when MN and TD increase. LME always decreases with the increase of TD, and increases with MN, while LDF increases with TD when MN is relatively large. This is because that when TD increases, the degree of vertex in derived edge-dependent graph also increases greatly. Hence, the size of MIS of the derived graph in each round will decrease. As a result, the rounds of the LDF will rise. To summary, the LME runs faster when TD is relatively lower, and LDF does better when TD is relatively higher.

In addition, we change the number of deployed nodes to check the scalability of the algorithms running times. From Fig. 14, we can see that our two algorithms both increase slow with network scale and has good scalability about times complexity.

## 6.5 More Realistic Connectivity Models

In the simulation, though the result is given in the UDG models, it is sufficient to demonstrate the necessity of optimization the selection of monitoring nodes. Notice that our algorithms are just based on the connectivity informations and work for more general graph model with theory guarantee.

## 7. DISCUSSIONS

### 7.1 Practical Applications

Since continuous monitoring of each node in the network may incur high energy cost, for energy-constrained WSNs, we may implement the self-monitoring with other energy conservation strategy such as scheduling, adaptive duty cycle, etc. Our goal to provide an underlying infrastructure for the upper layer protocols. In practical applications, we can take many manners to save the energy without the expense of the quality of self-monitoring. For example, each node in the network can schedule itself to switch into monitoring state as fixed or random periods. The nodes may

also be scheduled by the upper protocols using the local monitoring function. As such, the self-monitoring capability in the network can be on demand. When the network is in a lower security alert, each node may be in monitoring state in a shorter time; while when the network has a much higher security alert or requirements, more nodes can carry out monitoring operations.

The self-monitoring topology can be tailored to cater the different requirements and offer the fundamental assurance to other techniques using local monitoring, and can be implemented flexibly in practice.

### 7.2 Interference

In the real wireless network, interference may affect the selection strategy greatly, because collisions in the wireless channel can cause a high error rate for overhearing. If two links transmit data in the same time, the node that monitors the two links may overhear nothing due to the interference. Nevertheless, this does not affect our algorithms. There are several methods to deal with this situation. For example, we can evaluate and calculate and the interference level of candidate nodes, and restrict that a node with higher interference level being excluded during the preprocess stages for construction of self-monitoring topology. Only the nodes with a lower interference level are considered as candidates.

Indeed, interference has the *stochastic* feature and the level could be changed over time. If a node potentially suffers interference in some time slots, it still can monitor the links in other time slots.

### 7.3 Security

Security is another hinged issue for practical applications. The selection of patching monitoring nodes should be able to withstand a hostile attack against itself. We notice that we cannot eliminate the possibility of selecting malicious nodes, but we can manage to have the malicious nodes without higher possibility to become a patching monitoring node. The key problem is to validate the monitoring degree of a node which itself claims to have. This can be disposed by some cryptographic mechanisms, for example, authenticate the neighbor relation of a node.

### 7.4 Dynamic Maintenance

We assume that the active nodes in the network topology keep unchanged in most of the time. Certainly, after the generation of a topology, the active working topology might be subject to change due to node failures, etc. Our self-monitoring topology can also be updated correspondingly. If there is no major topology change, no update needs to be launched until some preset timer expires. On the other hand, for some major topology changes, an on-demand update can be performed. Notice that since our algorithms are completely distributed, the update process can be performed only in a local area where the change occurs.

### 7.5 Bounds for Approximation Algorithm

If additional information, such as location, power, etc, are available, more efficient heuristic algorithms for node selection can be obtained. An interesting question is that how to design constant-factor approximation algorithm for the MPMSP without using node location information. Our two algorithms set a good tradeoff between the performance (the theoretic bound of approximation algorithm) and the complexity of implementation (simple to run).

## 8. CONCLUSIONS

We address the issue of the optimal topology integrated with self-monitoring capability to meet the local monitoring requirements. We present a formal study on the problem of adding the minimum number of monitoring nodes into a large scale sensor network, and show that the problem is NP-complete. We also provide the upper bounds on the approximation ratio in a centralized scenario. We further design two distributed algorithms with provable approximation ratio and time complexity guarantee. Through comprehensive simulations, we present the performance comparison of our distributed algorithms and show that the optimized selection of monitoring nodes can decrease the number of monitoring nodes significantly.

Future work will involve the followings. First is to design secure node selection protocols to prevent malicious nodes from joining set of patching monitoring nodes and colluding. Second, we will investigate how real radio models can affect the selection strategy of monitoring nodes.

## 9. ACKNOWLEDGMENTS

This work is supported in part by the National Basic Research Program of China (973 Program) under grant No. 2006CB303000, the National High Technology Research and Development Program of China (863 Program) under grant No.2007AA01Z177 and 2007AA01Z180, NSFC Key Project grants No. 60533110 and 60736016, the Hong Kong RGC grant HKUST6169/07E, HKUST Nansha Research Fund NRC06/07.EG01, and Nokia APAC research grant.

## 10. REFERENCES

- [1] A. Perrig, "SPINS: security protocols for sensor networks," In Proc. of ACM MobiCom, 2001.
- [2] C. Karlof and D. Wagner, "Secure routing in wireless sensor networks: attacks and countermeasures," *Elsevier AdHoc Networks*, vol. 1, 2003.
- [3] H. Chan and A. Perrig, "Security and privacy in sensor networks," in *IEEE Computer*, vol. 36, 2003, pp. 103-105.
- [4] I. Khalil, S. Bagchi, and C. Nina-Rotaru, "DICAS: detection, diagnosis and isolation of control attacks in sensor networks," In Proc. of IEEE SecureComm, 2005.
- [5] I. Khalil, S. Bagchi, and N. Shroff, "LITEWORP: a light-weight countermeasure for the wormhole attack in multihop wireless networks," In Proc. of IEEE/IFIP DSN, 2005.
- [6] S. Ganeriwal and M. B. Srivastava, "Reputation-based framework for high integrity sensor networks," In Proc. of ACM SASN, 2004.
- [7] S.-B. Lee and Y.-H. Choi, "A resilient packet-forwarding scheme against maliciously packet-dropping nodes in sensor networks," In Proc. of ACM SASN, 2006.
- [8] A. Silva, M. Martins, B. Rocha, A. Loureiro, L. Ruiz, and H. Wong, "Decentralized intrusion detection in wireless sensor networks," In Proc. of ACM IWQoS, 2005.
- [9] K. Ioannis, T. Dimitriou, and F. C. Freiling, "Towards intrusion detection in wireless sensor networks," In Proc. of the 13th European Wireless Conference, 2007.
- [10] S. Marti, T. J. Giuli, K. Lai, and M. Baker, "Mitigating routing misbehavior in mobile ad hoc networks," In Proc. of ACM MobiCom, 2000.
- [11] S. Buchegger and J.-Y. L. Boudec, "Performance analysis of the CONFIDANT protocol: cooperation of nodes fairness in distributed ad-hoc networks," In Proc. of ACM MobiHoc, 2002.
- [12] Y. Huang and W. Lee, "A cooperative intrusion detection system for ad hoc networks," In Proc. of ACM SASN, 2003.
- [13] P. Michiardi and R. Molva, "CORE: a collaborativereputation mechanism to enforce node cooperation in mobile ad hoc networks," In Proc. of the IFIP Sixth Joint Working Conference on Communications and Multimedia Security, 2002.
- [14] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," In Proc. of the 33rd Hawaii International Conference on System Security, 2000.
- [15] B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris, "Span: An energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks," In Proc. of ACM MobiCom, 2001.
- [16] C. Schurgers, V. Tsatsis, S. Ganeriwal, and M. Srivastava, "Topology management for sensor networks: exploiting latency and density," In Proc. of ACM MobiHoc, 2002.
- [17] I. Khalil, S. Bagchi, and N. B. Shroff, "SLAM: sleep-wake aware local monitoring in sensor networks," In Proc. of IEEE/IFIP DSN, 2007.
- [18] C. Hsin and M. Liu, "Self-monitoring of wireless sensor networks," *Elsevier Computer Communications*, vol. 29, pp. 462-476, 2006.
- [19] F. Kuhn, R. Wattenhofer, and A. Zollinger, "AdHoc networks beyond unit disk graphs," In Proc. of ACM DIALM-POMC, 2003.
- [20] B. N. Clark, C. J. Colbourn, and D. S. Johnson, "Unit disk graphs," *Discrete Mathematics*, vol. 86, pp. 165-177, 1990.
- [21] M. R. Garey and D. S. Johnson, "The rectilinear Steiner tree problem is NP-complete," *SIAM Journal on Applied Mathematics*, vol. 32, pp. 826-834, 1977.
- [22] R. Tamassia and I. G. Tollis, "Planar grid embedding in linear time," *IEEE Transactions on Circuits and Systems*, vol. 36, pp. 1230-1234, 1989.
- [23] L. G. Valiant, "Universality considerations in VLSI circuits," *IEEE Transaction on Computers*, vol. C-30, pp. 135-140, 1981.
- [24] S. Schmid and R. Wattenhofer, "Algorithmic models for sensor networks," In Proc. of the 14th International Workshop on Parallel and Distributed Real-Time Systems, 2006.
- [25] A. Schrijver, *Combinatorial optimization - polyhedra and efficiency (Part III)*: Springer, 2003.
- [26] D. S. Hochbaum, *Approximation algorithms for NP-hard problems (Chapter 3, page:100-102)*: PWS Publishing Company, 1997.
- [27] D. S. Hochbaum and W. Maass, "Approximation schemes for covering and packing problems," *Journal of the ACM*, vol. 32, pp. 130-136, 1985.
- [28] P.-J. Wan, K. M. Alzoubi, and O. Frieder, "Distributed construction of connected dominating set in wireless ad Hoc networks," In Proc. of IEEE INFOCOM, 2002.
- [29] B. Liang and Z. J. Haas, "Virtual backbone generation and maintenance in ad hoc network mobility management," In Proc. of IEEE INFOCOM, 2000.
- [30] D. S. Hochbaum, *Approximation algorithms for NP-hard problems (Chapter 3, page:109-111)*: PWS Publishing Company, 1997.