CitySee: Urban CO_2 Monitoring with Sensors

Xufei Mao*†, Xin Miao†, Yuan He*†, Xiang-Yang Li*‡, and Yunhao Liu*†

* MOE Key Lab for Information System Security, School of Software, TNLIST, Tsinghua University, China
† Department of Computer Science and Engineering, HKUST
‡Computer Science Department, Illinois Institute of Technology, USA
Email: {xufei.mao, yunhao}@greenorbs.com, {miao, heyuan}@cse.ust.hk, xli@cs.iit.edu

Abstract—**Motivated by the needs of precise carbon emission measurement and real-time surveillance for CO_2 management in cities, we present CitySee, a real-time CO_2-monitoring system using sensor networks for an urban area (around 100 square kilometers). In order to conduct environment monitoring in a real-time and long-term manner, CitySee has to address the following challenges, including *sensor deployment*, *data collection*, *data processing*, and *network management*. In this discussion, we mainly focus on the sensor deployment problem so that necessary requirements like connectivity, coverage, data representability are satisfied. We also briefly go through the solutions for the remaining challenges. In CitySee, the sensor deployment problem can be abstracted as a relay node placement problem under hole-constraint. By carefully taking all constraints and real deployment situations into account, we propose efficient and effective approaches and prove that our scheme uses additional relay nodes at most twice of the minimum. We evaluate the performance of our approach through extensive simulations resembling realistic deployment. The results show that our approach outperforms previous strategies. We successfully apply this design into CitySee, a large-scale wireless sensor network consisting of 1096 relay nodes and 100 sensor nodes in Wuxi City, China.**

Index Terms—**CO_2 monitoring, relay nodes placement, wireless sensor networks.**

I. INTRODUCTION

With the worsening of Global Warming, the carbon sequestration and emission issues have attracted serious concern of researchers and specialists all over the world. One of the main causes deteriorating global climate is the over-emission of CO_2. Most countries, especially massive energy-consuming countries are required to reduce (or restrict) the emission of CO_2 in order to slow down the steps of Global Warming. Since arguably more than 80% of CO_2 emissions originate in urban areas, which occupy less 2.4% of land mass globally [11], understanding the relationships between the form and pattern of urban development and the carbon cycle is crucial for estimating future trajectories of greenhouse gas concentrations in the atmosphere and facilitates mitigation of climate change. Hence, accurately measuring the emissions of CO_2 of the interested (or serious) areas is in great demands, especially for those high CO_2 emission areas, like industrial regions, densely populated places, and etc.

One of the methods estimating CO_2 emissions is based on raw material (like fossil oil and coal) consumption, which is widely used by European Environment Agency when it ranks countries with their CO_2 emissions every year [2]. This is reasonable to some extent but less of accuracy and fairness since different countries or cities have different carbon sequestration abilities, which is the opposite of carbon emissions. Another inference-based carbon sensing technology is adopted to measure CO_2 fluxes [8]. When a radiation (e.g., a thermal infrared radiation) travels through the atmosphere, it can be either absorbed or re-emitted. CO_2 is one of the gases which are responsible for most of the absorption. Consequently, by measuring the radiation above the atmosphere, CO_2 fluxes could be inferred. Estimation-based and inference-based methods can satisfy the large-scale and real-time requirements for CO_2-monitoring with the cost of sacrificing accuracy. There are some expensive instruments (*e.g.*, HD-CO2-S [1]) in the market, which are able to measure the CO_2 fluxes for a location during a fixed time period. However, it is very hard to use them for long-term measurement of CO_2 emissions due to the constraints like high cost and requiring electronic power support. Indeed, we still lack effective approaches to measure the carbon emissions of large-scale areas accurately and thoroughly in a real-time and long-term manner.

Fortunately, the combination of sensors with mature technology and small individual wireless node show great potential for us to interact with the physical world. On the one hand, the manufacturing of small and low cost sensors and wireless terminals becomes technically and economically feasible. On the other hand, the technologies (like routing protocols, link scheduling protocols, power consumption) of wireless sensor networks (WSNs) become more and more mature. Large-scale, self-organized WSNs consisting of thousands of individual nodes have been applied in real application scenarios, like GreenOrbs [21] and ExScal [4].

The goal of *CitySee* is to deploy thousands of wireless sensor nodes in an urban area of Wuxi City, China, such that multi-dimensional data including CO_2, temperature, humidity, light, location, and etc. could be collected in a real-time manner for further analysis. Compared with GreenOrbs [21] in the forest environment, the long-term, large-scale, continuous, and synchronized surveillance of huge measurement in urban area is more difficult due to complex geographical environment. The following four issues should be properly solved for CitySee.

1) *Sensor deployment*: According to the requirement from ecologists, the wireless nodes equipped with different types of sensors should be placed in interesting points

(a) SNs. (b) RNs.

Fig. 1. Two types of wireless nodes. Each SN is equipped with a CO_2 sensor and each RN has one temperature/humidity sensor and one light sensor on it.

inside the monitoring area such that all collected data are representative. More importantly, some relay nodes will be further inserted such that all individual nodes could form a connected WSN.

2) *Data collection*: An agency should be planted into each sensor node in order to sense the environment, package data and send data back to the sink node (may need nodes to work collaboratively).
3) *Data processing*: At the base station, all collected data are recorded, formatted, and displayed.
4) *Network diagnosis management*: To keep the entire network run, system administrators needs efficient tools to understand the running status, locate the error if any, and further recover from failures.

In this discussion, we focus more on the first issue: how to deploy a large-scale WSN consisting of different types of nodes into the monitoring area for real-time collection of multi-dimensional environment information in CitySee. Basically, we design two types of nodes equipped with CO_2 sensors and other type of sensors (*e.g.*, temperature, humidity, and light) respectively. Due to the fact that the sensing operation of a CO_2 sensor is very energy-consuming, a wireless node equipped with a CO_2 sensor (abbreviated as SN) does not relay packets for any other nodes while the other type of nodes perform both sensing and relaying operations (abbreviated to RN). Please refer to Fig. 1 for illustration. In addition, some special locations where SNs and RNs have to be placed exactly in the area have been chosen *a priori* by ecologists to ensure that collected data are representative. At the same time, some locations (*e.g.*, buildings) are strictly prohibited to deploy any node due to physical constraints.

When performing deployment, one major benchmark is the connectivity of the network. The phase one of our Carbon Monitoring Project, GreenOrbs [21], is successfully deployed in the forest, which aims at evaluating carbon sequestration ability (an opposite of carbon emissions). The placement for GreenOrbs which aims at evaluating carbon sequestration ability is easier than that for CitySee since the density of obstacles in forests is generally more uniform than that in cities. Through field measurement of communication links among wireless nodes, we found that there are many "black holes" (*e.g.*, buildings) in the urban area, which can "absorb" the wireless signals by blocking, reflecting, interference, and etc. CitySee, must minimize the total number of RNs for maintaining the connectivity of the deployed network from both economic and applicable concerns after the critical locations of a subset of nodes have been chosen. Our solution should also satisfy the following constraints all the time, 1) a SN does not relay data packets for any other node; 2) some locations are not available for deploying RNs; 3) wireless links should avoid "holes".

The major contributions of this paper are as follows.

1) We report the design and implementation of CitySee, a large-scale (100 SNs and 1096 RNs) and long-term CO_2 monitoring WSN in urban area, Wuxi City, China, which is a complementary project of GreenOrbs.
2) We propose low-cost sensor deployment strategies with guaranteed performance (within 2-approximation ratio of the optimum) which addresses the sensor deployment problem in CitySee.
3) To the best of our knowledge, we are the first to define geometric version of Group Steiner Tree with Holes problem, which is very helpful to a rich body of sensor deployment problems by considering both the optimization objectives of sensor deployment problems and realistic deployment constraints.
4) Performance of our proposed approaches has been extensively evaluated through simulations resembling real deployments.

The rest of the paper is organized as follows. We first review related works in Section II. In Section III, we present our network model and formally define the problems to be studied in this paper. We divide the relay nodes deployment problem into several subproblems and propose solutions to them in Section IV. We examine the performance of our approaches in Section V. The procedure of remaining three steps of CitySee are presented briefly in Section VI. We conclude the work in Section VII.

II. RELATED WORK

Generally speaking, previous work on relay nodes placement according to connectivity can be classified into two categories, *i.e.*, single-tiered node placement and two-tiered node placement.

Single-tiered placement assumes that both SNs and RNs participate in forwarding packets received from other nodes. Lin and Xue defined it as a *Steiner minimum tree with minimum number of Steiner points and bounded edge length* problem in [19]. They proved its NP-hardness and proposed a 5-approximation algorithm based on minimum spanning trees. Chen *et al.* [9] studied the same problem and presented a 3-approximation solution. In [10], Cheng *et al.* developed a faster 3-approximation algorithm as well as a randomized 2.5-approximation algorithm. Lloyd *et al.* [23] assumed relay nodes may have a larger communication range than sensor nodes and presented a 7-approximation algorithm.

In two-tiered node placement, only RNs are able to forward packets received from other nodes while SNs only send packets generated by themselves to some RN(s). In [17], Hao *et al.* designed an $O(D \log n)$-approximation algorithm for 2-connectivity and distinct communication ranges, where n is the network size and D depends on network diameter. Based on the assumption that relay nodes' communication range is at least four times as large as sensor nodes' communication range, Tang *et al.* [27] provided 6-approximation and 4.5-approximation algorithms for 1-connectivity and 2-connectivity cases respectively. Further results can also be found in [23]. Our work belongs to two-tiered node placement problem with the constraint that there is a set of unavailable locations for deploying RNs.

A bunch of work studied sensor deployment problem from other points of view. For example, [5] proposed a strip-based deployment pattern ensuring both coverage and 2-connectivity. [14] considered the joint optimization of sensor placement and topology construction subject to constraints on the distortion of reconstructed sensing data and network lifetime. In addition, [24] presented a random deployment approach to achieve minimum total energy. Some other works [3], [6] etc. mainly studied data collection problem using a WSN with mobile sink (data mule). Their work focused on data transmission related strategies such as to improve the throughput, reduce the energy consumption and delay. The work in [29] concentrated on capacity and delay tradeoffs for the data collection (convergecast) scenario. [7], [30] studied sensor deployment problem from coverage aspect. In monitoring applications, it is often required that each point of the interesting area is covered by at least 1 or k sensors.

Compared to the previous work, the main novelty of our problem is that we study constrained two-tiered relay nodes placement problem to meet connectivity requirement. We consider the joint optimization of sensor placement in an urban area with complicated physical constraints. The large area of required monitored region, complicated physical constraints and the consequent scale of wireless sensor networks render our problem more challenging.

III. PROBLEM FORMULATION

A. Network Model

Given two types of wireless sensor nodes RNs and SNs with same transmission range r, we assume that two location (point) sets $PS = \{p_1, p_2, \cdots, p_m\}$ and $LS = \{l_1, l_2, \cdots, l_n\}$) where SNs and RNs should be placed are chosen *a priori* in the two-dimensional map of the deployment region Ω. For simplicity, we use u to denote the location of the node u if no confusion arises. We use $d(u, v)$ to denote the Euclidean distance between two points (nodes) u and v in the plane and use $d(u \sim v)$ to indicate the Euclidean distance of the path connecting nodes u and v. In addition, there are some places in reality where we cannot deploy wireless nodes directly due to physical constraints, *e.g.*, some buildings, we call these kind of places **holes** in the map. Assume there are totally p holes in region Ω, saying in set $\mathscr{H} = \{h_1, h_2, \cdots, h_p\}$. We further assume that each hole h_i is a polygon with constant number of edges and the shapes and positions of all holes in region Ω are known. Since a wireless link between two wireless nodes either is unavailable or has very low transmission rate when there are some explict obstacles between them, we consider that any wireless link going into or across a hole is unavailable.

Regulation 1: A wireless link from u to v exists iff 1) v falls into the transmission range of u; 2) v is a RN; and 3) the line from u to v does not go into or across any hole.

We use the following standard graph theoretic notations: for a graph G, $V(G)$ denotes the vertex set of G and $E(G)$ denotes the edge set of G. Since we would like to deploy sensor nodes on some area with *holes*, from now on, when we say something under "hole-constraint", we mean something avoids the holes on the deploy area.

B. Questions Studied

We give the formal definition of the question studied in this paper.

Question 1: Given the two-dimensional map of deployment region Ω with a known hole set $\mathscr{H} = \{h_1, h_2, \cdots, h_p\}$, assuming that m SNs and n RNs have already been placed at location sets $PS = \{p_1, p_2, \cdots, p_m\}$ and $LS = \{l_1, l_2, \cdots, l_n\}$ respectively, the question is how to further deploy minimum number of RNs in Ω such that all deployed SNs and RNs construct a connected WSN following Regulation 1. Here, "connected" means that any SN can send packets to at least one RN and any two RNs can reach each other through some path using RNs only.

After deploying both SNs and RNs following the locations sets PS and LS, there may be some isolated components. In order to using fewer RNs to connect these isolated components into a connected WSN, we need to find some shortest paths under hole-constraint to connect at least one node from each component. Hence, our problem is transferred into geometric version of Group Steiner Tree problem [16] under hole-constraint (defined in Sub-Question 2) by assuming that nodes belonging to the same component form a group. Then, we deploy RNs along the shortest paths we got, we finish our work. However, this may be not always the truth since a SN will not relay data packet for any other node such that a regularly defined component we got after deploying SNs and RNs following PS and LS respectively is actually not connected. For instance, case (a), (b) and (c) in Fig. 2 are not "legally" connected indeed since Regulation 1 is broken.

Clearly, the precondition of solving our problem using aforementioned idea is to find all self-connected components following Regulation 1. Next, we split Question 1 into two sub-questions. The first one is to find all legal components (defined as Sub-Question 1) and the other one is geometric Group Steiner Tree with Holes problem (defined as Sub-Question 2). Here, the output of Sub-Question 1 will be the input of Sub-Question 2.

Sub-Question 1: Given the two-dimensional deployment region Ω with known hole set $\mathscr{H} = \{h_1, h_2, \cdots, h_p\}$, assuming that m SNs and n RNs have already been placed at location

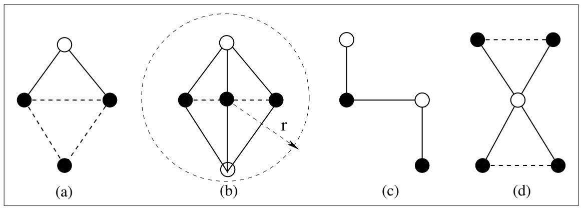


Fig. 2. (a), (b) and (c) are illegal components and case (d) is legal. Black and white nodes denote SNs and RNs respectively with transmission range r. The solid line segments denote communication links between wireless nodes while dotted line segments are not considered as valid communication links in our case.

sets $PS = \{p_1, p_2, \cdots, p_m\}$ and $LS = \{l_1, l_2, \cdots, l_n\}$ respectively, the question is how to divide all deployed nodes into self-connected components where each wireless link inside a component satisfies Regulation 1.

Sub-Question 2: **Geometric Group Steiner Tree with Holes (G-GSTWH)**: Given the two-dimensional map of deployment region Ω with known hole set $\mathscr{H} = \{h_1, h_2, \cdots, h_p\}$, assuming that there is a collection of groups $G_1, G_2, \cdots, G_d$ where each group contains a bunch of nodes and $G_i \cap G_j = \phi, 1 \le i \ne j \le d$, the problem is to construct a minimum edge-weighted (Euclidean distance) tree under hole-constraint, which spans at least one node from each group.

Obviously, if we find the solution of G-GSTWH, a Steiner tree under hole-constraint connecting all groups, we can replace edges of this Steiner tree with the connected communication paths consisting of RNs. Note that, some component $S_i \in \mathscr{S}$ may be not considered in Sub-Question 2 if S_i contains one SN only. For each isolated SN, finding a shortest path under hole-constraint connecting it to its closest RN, we've done. The reason for us to do this is because there are few isolated SNs in our case since a location in PS is always surrounded by several locations in LS for CO_2 accurate fluxes measurement purpose. When there are large number of isolated SNs, the question comes to terminal steiner tree problem, which was solved in [12].

IV. SOLUTIONS

The main idea to solve Sub-Question 1 is as follows. Basically, what we need is to find all components under hole-constraint among all deployed RNs whose locations correspond to LS without considering any SN. Then we obtain all legal components by adding each SN to some resultant component obeying Regulation 1 if possible. It is not difficult to show that we can obtain all components in $O(n^2p+m)$ time where m and n are cardinalities of PS and LS respectively, p is the number of holes in region Ω. (Please refer to [25] for detailed proof.)

After we obtain all legal components (assuming in set $\mathscr{S}$), the next step is to deploy minimum number of RNs to connect all pre-deployed SNs and RNs. Assuming that the component set we got is set$\mathscr{S} = \{S_1, S_2, \cdots, S_d\}$ where $S_i = (V(S_i), E(S_i))$ is the i^{th} component. Here, $V(S_i)$ contains all sensor nodes belonging to component S_i and $E(S_i)$ contains all possible links (obeying Regulation 1) among nodes in $V(S_i)$. For instance, in Fig. 3(a), the shadowed area are obstacles (buildings in our case) where we cannot deploy sensor nodes (at least in two-dimensional space). In addition, two sensor nodes within the transmission range of each other may not communicate or with very low transmission ratio due to holes so that this kind of links are unavailable.

Next, we obtain a new group set $\mathscr{G} = \{G_1, G_2, \cdots, G_e\}$ by removing all SNs and corresponding edges, i.e., $G_i = (V(G_i), E(G_i))$, $V(G_i) = V(S_i) \setminus \{\text{all SNs}\}$ and $E(G_i) = E(C_i) \setminus \{\text{all edges incident on SNs}\}$. Clearly, if we are able to find a feasible solution to connect all nodes in group set $\mathscr{G}$, the same solution can be applied to connect all nodes in component set $\mathscr{S}$ since no SNs will be chosen as the bridge point connecting two components.

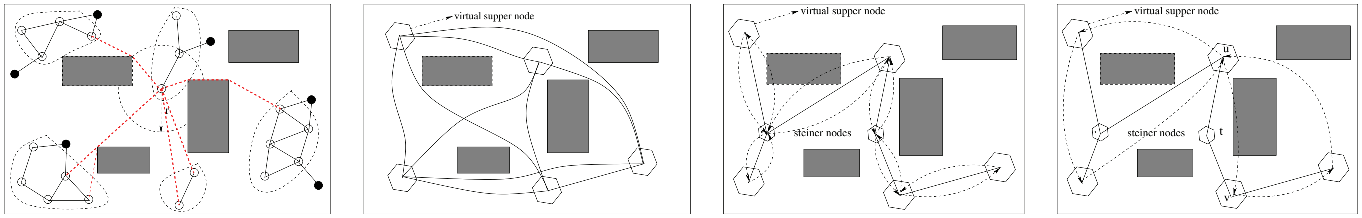
Now we are ready to solve G-GSTWH problem (Sub-Question 2) using group set $\mathscr{G}\{g_1, g_2, \cdots, g_k\}$. As we have mentioned before, in CitySee, the number of isolated components containing single CO_2 are very small since any required location for a SN is always close to some required locations for several RNs in order to improve the accuracy of computing CO_2 fluxes in the air. From now on, we concentrate on the case that the number of isolated components containing single CO_2 node is within small fraction of all CO_2 nodes.

Our main idea is as follows. We first use method in [18] to find the shortest path under hole-constraint between each pair of nodes u and v in different groups, *i.e.*, $u \in G_i, v \in G_j$ where $i \ne j$. Here, if the shortest path under hole-constraint connecting u and v is a series of continuous line segments (starting from u and ending at v), the Euclidean length of shortest path under hole-constraint between u and v is the summation of Euclidean length of each line segment. Next, we consider each group as a virtual supper node and the shortest path under hole-constraint connecting two virtual supper nodes is defined as the shortest path connecting two sensor nodes from each of two virtual supper nodes (groups), *i.e.*, the Euclidean length of the shortest path connecting two virtual supper nodes G_i and G_j is $d(G_i \sim G_j) = \min\{d(u,v) : \forall u \in G_i, \forall v \in G_j\}$. Hence, we have a complete graph $G_{grp} = (V(G_{grp}), E(G_{grp}))$ where $V(G_{grp})$ contains all virtual supper nodes and $(G_i, G_j) \in E(G_{grp})$ is the shortest path connecting (G_i, G_j) under hole-constraint (Fig. 3(b)).

By using a minimum Euclidean spanning tree (MST) method (*e.g.*, Prim's algorithm), we can find a MST to connect all groups. Next, we replace each edge of resultant MST with original continuous line segments. Based on all resultant paths(line segments) connecting all components, we deploy more RNs along these paths where each pair of adjacent nodes are within r. Hence, we have a connected WSN. The details of our algorithm please refer to Algorithm 1.

Lemma 1: The Algorithm 1 has a 2-approximation ratio of the optimum of Sub-Question 2 with time complexity $O(n + m + p^2 \log n + p^2 \log m)$ where m and n are the cardinalities of PS and LS respectively, p is the number of holes.

Proof: Given group set $\mathscr{G}$, in which each group is a virtual supper node, we show that a MST under hole-constraint spanning all virtual supper node has total length as most twice



(a) (b) (c) (d)

Fig. 3. (a) Some legal components in the original graph. White nodes and Black nodes denote RNs and SNs respectively. All nodes cycled by dotted curve belong to the same component. The dotted red lines show some shortest path connecting RNs in different components. (b) Complete graph of all virtual super nodes denoting by hexagons. (c) Euler tour spanning all virtual super nodes. (d) Hamiltonian cycle spanning all virtual super nodes. The shadow areas are holes and smaller hexagons indicate steiner nodes.

Algorithm 1 Shortest Paths Connecting All Components

```
Input: Given group set $\mathscr{G} = \{G_1, G_2, \cdots, G_d\}$ and map with known hole set $\mathscr{H}$
Output: Shortest Paths connecting all groups under hole-constraint.
for each pair of nodes $u$ and $v$ where $u \in G_i, v \in G_j : 1 \le i \ne j \le d$ do
    Use method in [18] to compute the shortest Euclidean path between $u$ and $v$ under hole-constraint
Assuming each group in set $\mathscr{G}$ is a virtual supper node
for Each pair of virtual supper nodes $G_i$ and $G_j$ in $\mathscr{G}$ where $i \ne j$ do
    Compute the minimum Euclidean path under hole-constraint between $G_i$ and $G_j$, i.e., $d(G_i \sim G_j) = \min\{d(u,v) : \forall u \in G_i, \forall v \in G_j\}$
Using Prim's algorithm to find a MST under hole-constraint to connect all virtual supper nodes.
for Each component which contains one single $CO_2$ node only do
    Find a shortest path under hole-constraint connecting itself to the nearest RNs (maybe use some new deployed RNs).
```

of the optimum, *i.e.*, the total length of an minimum edge-weighted group Stein Tree spanning all virtual nodes.

Assuming the optimum Steiner tree under hole-constraint with the minimum total length has cost OPT. By doubling its edges we obtain an Eulerian graph connecting all virtual super nodes and, possibly, some Steiner vertices. Next, we find an Euler tour of this graph, for instance by traversing the edges in depth first search order, see Fig. 3(c) for illustration. Clearly, the cost of this Euler tour is $2OPT$. Then, we obtain a Hamiltonian cycle on the vertices (all super nodes and steiner nodes) by traversing the Euler tour and"short-cutting" Steiner vertices and previously visited vertices (all super nodes). By removing an edge from this cycle, we obtain a path that spans $\mathscr{G}$. Noticing that, when we short-cut Steiner vertices during traversing, we use the shortest path under hole-constraint to connect two vertices. For example, in Fig. 3(d), for two virtual supper nodes u and v, the OPT solution use two line segments (u,t) and (t,v), which will be replaced by $(u \sim v)$ (the shortest path connecting u and v) in the Hamiltonian cycle. Since $(u \sim v)$ is the shortest path connecting u and v under hole-constraint, the length of $(u \sim t \sim v)$ is no smaller than $(u \sim v)$, then the triangle inequality still holds such that the path spanning $\mathscr{S}$ has cost at most $2 \times OPT$. Remembering that this path under hole-constraint is also a spanning tree on $\mathscr{S}$, the MST under hole-constraint has cost at most $2 \times OPT$.

Next, we prove the time complexity. First, the cardinality of component set $\mathscr{S}$ and resultant group set $\mathscr{G}$ is at most n. Next, computing the shortest path between each pair of nodes in different groups takes $O(n + p^2 \log n)$ time [18]. Clearly, computing the shortest distance under hole-constraint between each pair of virtual super nodes will take $O(n^2)$ time since we at most have n groups. Using Prim's algorithm to find a MST among all groups takes us $O(n^2 \log n)$ time since the number of edges of the complete graph among all groups is $O(n^2)$. Finally, to find a shortest path connecting each isolate CO_2 to its nearest RN is take $O(m + p^2 \log m)$ by [18]. Hence, the total running time is $O(n + m + p^2 \log n + p^2 \log m)$. ■

V. EXPERIMENTS

A. Deployment of RNs in CitySee

We successfully apply the solution for RNs replacement to CitySee using the G-GSTWH-based algorithm. In CitySee, location sets PS and LS with cardinality 100 and 400 respectively are given *a priori* in the required deployment region. By the G-GSTWH-based algorithm, we further deploy 696 RNs to connect all SNs and RNs whose locations are corresponding to PS and LS. We deploy both types of sensor nodes in trees and telegraph poles around 2.5 meters Euclidean distance to the ground. See Fig. 4 for illustration.

Part of deployment area and the resulting topology (obtained from real data trace) is shown in Fig. 5, from which we can see that the impact on valid wireless links by holes are obvious. For instance, we have some valid wireless links with Euclidean distance around 150 meters in some open territories while some nodes do not construct valid links even they are close enough due to "holes".

B. Simulations

We conduct extensive simulations to verify the efficiency of our approaches. Besides the main G-GSTWH based algorithm, we also implement two baseline algorithms *Simple MST-based*

Fig. 4. RNs and SNs hanged on telegraph poles and trees.

Fig. 5. Partial deployment area and the resultant topology (a snapshot of real data trace). Red nodes and yellow nodes denotes SNs and RNs respectively.

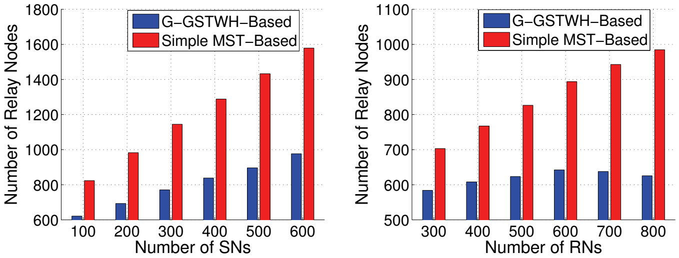
and *Random-based*. Simple MST-based strategy first places a RN at each location in PS such that every SN has at least one RN (at the same location) to help to relay data packets. It then constructs a minimum spanning tree under hole-constraint spanning all RNs whose locations are corresponding to the PS and LS sets. More RNs are further deployed along edges (of the MST) with length larger than r. Random-based strategy randomly sprays RNs into the monitored area Ω until each SN in PS and each RN in LS are able to find a path to the sink, which consists of RNs only. We implement all three algorithms in $C++$ and all simulations are performed in the same software environment on a Pentium R 2.7GHz machine.

For all test cases, we first randomly and uniformly generate a hole set in a rectangle region with area of 5000 meters × 4000 meters. Then, the location sets PS and LS are generated in this region without any hole i.i.d. After that, we investigate the number of RNs required for different algorithms by adjusting five parameters: (i) m: cardinality of location set PS; (ii) n: cardinality of location set LS; (iii) r: transmission range of a sensor node; (iv) p: number of holes; and (v) the size of holes. Unless specifically mentioned, the default values of parameters are shown in the Table I.

For each set of parameters, we run the simulation for 10 times. We do not list the results for random-based algorithms in charts since the average performance is almost 10 times worse than those of the other two algorithms. We concentrate on comparing G-GSTWH-based algorithm with simple MST-based algorithm. For different test cases, G-GSTWH-based algorithm has better performance (gain from 20% to 35%) than that of simple MST-based algorithm.

TABLE I
DEFAULT VALUES OF SIMULATION PARAMETERS.

Parameter	Default Values
Deployment Area	5000 (m) × 4000 (m)
Transmission Range	100 (m) for all nodes
Size of Holes	100 (m) × 80 (m)
p	50, randomly deployed
m	100, randomly deployed
n	500, randomly deployed



(a) m is variant. (b) n is variant.

Fig. 6. The simulation results when we adjust m, n respectively where fixing other parameters. The size of a hole size is 100 × 80.

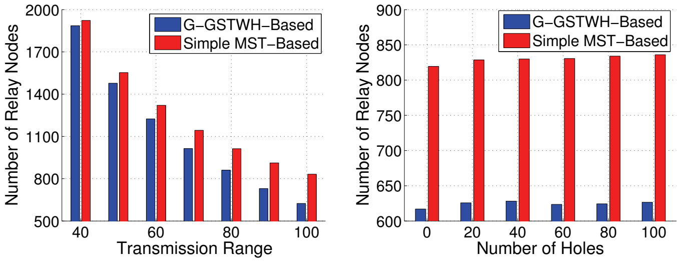
C. Impact of SN Density and RN Density

In the first set of simulation, we increase m (the number of pre-deployed SNs) from 100 to 600 with step 100 while other parameters use default values. As we can see from Fig. 6(a), the G-GSTWH-based algorithm outperforms single MST-based algorithm. Although the total number of new deployed RNs for two algorithms increases with the increment of m, the increment trend of G-GSTWH-based algorithm is slower.

In the second set of simulation, we increase n from 500 to 800 with step 50 while fixing other parameters. An interesting phenomena is that when the number of pre-deployed RNs is larger than 600, the total number of needed RNs for the G-GSTWH-based algorithm decreases while that of simple MST-based algorithm continues to increase. We analyze the reason and realize that the number of "legal" components obtained by our algorithm is stable in this case such that the larger n is, the closer components are, which causes different trends of two algorithms with the increment of n. Please refer to Fig. 6(b) for illustration.

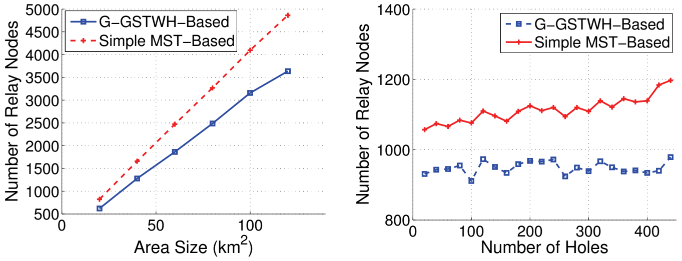
D. Impact of transmission range r and the number of holes p

In the third set of experiment, we first increase r from 40 to 100 with step 10 while other parameters use default values. The results for both algorithms are shown in Fig. 7(a), the decreasing trend of G-GSTWH-based algorithm is faster that of simple MST-based algorithm. In order to verify the impact caused by the number of holes, we run all algorithms by increasing the p from 0 to 100 with step 10. Surprisingly, the



(a) r is variant. (b) p is variant.

Fig. 7. The simulation results when we adjust r, p respectively where fixing other parameters.



(a) area scales. (b) p scales.

Fig. 8. # of nodes needed when system scales.

total number of new deployed RNs only has slight increment (Fig. 7(b)). We conjecture that the impact of scaling p becomes smaller when r is close to the edge length of holes. Actually, this is verified by the simulation results later.

E. Scalability

In the forth set of simulation, we scale both the area of deployment region and the number of holes p while fixing other parameters. We adjust the area of deployment region from 20 to 10240 square kilometers and vary the number of holes from 20 to 100 while keeping the density of m, n unchanged. For all these test cases, we randomly generate the shape (length and width) of holes. Both results shown in Fig. 8 and Fig. 9 indicate that for the G-GSTWH algorithm, when the transmission range of sensor nodes is around the edge length of holes, the increment of number of holes does small impact on the scalability. This is note true for the simple MST-based algorithm.

VI. RECENT ADVANCES

A. Hardware and Software

The hardware platform in CitySee is based on TelosB. Both SN and RN use MSP430F1611 processor and CC2420 radio such that they are able to exchange data with each other using ZigBee protocol in 2.4 GHz. In addition, each SN is equipped with CO_2 sensor while every RN has temperature/humidity sensor and light sensor on it. All nodes are encapsulated with industrial grade design in order to adapt to outdoor environment.

We develop software for different types of sensor nodes on top of TinyOS 2.1.1, which consists of the following major components. First, we implement the link estimation component using the four bit link estimation method [13] to regularly maintain a neighbor table. Second, we use the default Low Power Listening MAC protocol of TinyOS to reduce the energy consumption. Third, the multi-hop routing component is implemented based on the CTP [15] protocol for data collection.

Fourth, we apply the Drip protocol [28] to disseminate key system parameters in terms of the dissemination component, such as transmission power, sampling frequency, duty cycle and etc. In CitySee, a node is programed to sample the environment data according to different types of sensor in every 10 minutes, and then sends the data packet to the sink node through one- or multi-hops. In addition, a SN drops any data packet from other SNs or RNs.

B. Data Collection

Collection Tree Protocol CTP [15] is adopted for multi-hop sensing data collection. We collect three types of data packets, each of which is responsible for different types of information. The C_1 type packet contains two types of information: (1) sensing data, including temperature, humidity, light, or CO_2 concentration; and (2) routing information, including path-ETX [15] from the original packet-source to the sink node. Thus, we are able to obtain the complete routing path of any packet by piggybacking these information into sensed data. The C_2 type of packet records local information for each sensor node. Typically, a C_2 contains the routing table including IDs and RSSI values from its neighbors, the link-ETX estimation value of links to its neighbors. A type C_3 packet contains more detailed information inside a single wireless node. For instance, the CPU counter records the accumulated task execution time, the radio counter records the accumulated radio-on-time, the transmit counter records the accumulated number of transmitted packets, the receive counter records the accumulated number of received packets, and the loop counter records the accumulated number of detected loops.

In Fig. 10, each individual node is described as a circle whose area indicates the number of packets it has transmitted for the last 10 minutes, and all used wireless links are shown as well. Figure 11 displays the accumulated number of tasks each wireless node has executed in a 3D format based on the physical topology of the entire wireless sensor network. In CitySee, the traffic load of wireless sensor nodes with different roles (e.g., relay nodes, sensor nodes) are quite different depending on their physical environment and routing protocols. For example, the number of tasks executed by some node closed to the sink node could be up to 8742 times in 10 minutes while around $\frac{1}{4}$ of wireless nodes have the average number 2 only.

C. Data Processing

We have collected over 8 GB data traces from 1196 wireless sensor nodes including all environment-related data for the purpose of CO_2 emission analysis and network status-related

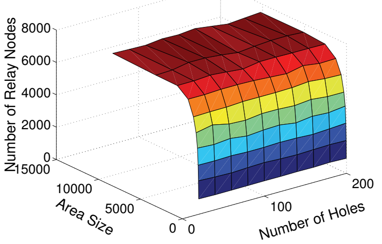


Fig. 9. Both area and p scale.

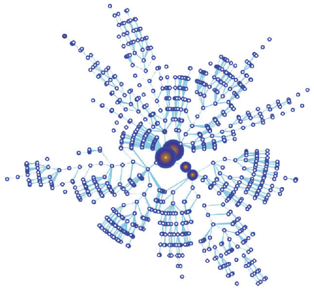


Fig. 10. Logical topo and traffic.

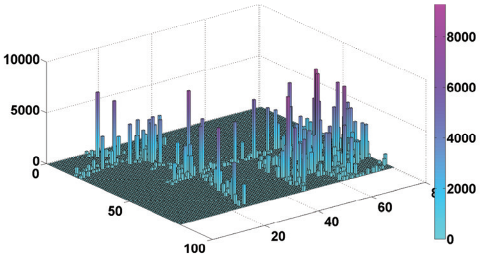


Fig. 11. Executed tasks of each node.

data for the purpose of network management and diagnosis. Combining all three types of data packets, we construct the entire network at the base station using the real map. The 2D geometric location of each node is obtained when it was deployed. Figure 12 is the snapshot (obtained from real data trace) of CitySee, in which white nodes denote wireless sensor nodes and the rectangles indicate the packets in the air. All links shown are the active links of last 10 minutes. The network diameter of the deployed network by hop count reaches nearly 35 and the longest hop distance observed in CitySee is 20 hops. In CitySee, the radio duty cycle is 4%. Upon all collected data, we design and implement visualization interfaces to depict the changing of environment. For instance, Fig. 13 shows the contour map of CO_2 concentration in the monitored area, in which the darker the color is, the higher CO_2 concentration is.

D. Network Management

Since CitySee has the long-term running objective and any physical modification of the network (e.g., replacing individual node) is pretty costly, it is critical and necessary to learn the running status of the entire network as well as each individual node. In order to collect key metrics (such as radio duty cycles, the number of packet transmissions and receptions) and provide visibility into the system, we further design and implement the network management and diagnosis component [20], [22], [26]. In Fig. 14, we show the node management interface by which network administrators are able to trace the node status easily, including its data transmission ratio, neighbor-related information (RSSI, LQI), and etc. Comprehensive consideration, we design several decades of indices to evaluate the healthy of CitySee, e.g., data reception ratio, the total number of tasks executed, routing loops detected, traffic analysis, and etc., which is shown in Fig. 15.

VII. CONCLUSIONS

We present CitySee, a CO_2-monitoring project using a large-scale wireless sensor network in a urban area in Wuxi, China. We focus on the solution of relay node placement problem, one of four major components of CitySee, by formulating the G-GSTWH problem and giving a 2-approximation ratio solution. There are many future works remaining. For example, if LS is empty such that we only consider to deploy some RNs to connect all deployed SNs under the hole-constraint, the question becomes terminal Steiner tree with holes. In addition, our solution only guarantee a 1-connected wireless sensor network. Can we have effective and efficient method to guarantee a k-connected wireless sensor network? Another interesting idea is to use WSNs to construct virtual Carbon Flux Towers, which are utilized to monitor carbon flux accurately but expensively in the real-time manner. However, constructing virtual Carbon Flux Towers raises many challenges at the same time, e.g., multiple vertical layers, 3-dimensional deployment, longer power-lasting requirement, and etc. We leave all these interesting issues for future study.

VIII. ACKNOWLEDGEMENT

This study is supported by the NSF China Major Program 61190110. We would like to thank other people who also contribute to CitySee, including Zhichao Cao, Dr. Guojun Dai, Wei Gong, Rui Li, Shuo Lian, Dr. Kebin Liu, Junliang Liu, Qiang Ma, Wei Xi, Dr. Panlong Yang, Dr. Zheng Yang, Lan Zhang, Dr. Jizhong Zhao, and etc.

REFERENCES

[1] co_2 detecting instruments. In *http://www.szken.com/product/6.html*.
[2] *eea*. In *http://www.eea.europa.eu/*.
[3] ANASTASI, G., CONTI, M., MONALDI, E., AND PASSARELLA, A. An adaptive data-transfer protocol for sensor networks with data mules. In *Proceedings of IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks* (2007), pp. 1–8.
[4] ARORA, A., RAMNATH, R., AND ET AL, E. E. Exscal: Elements of an extreme scale wireless sensor networks. In *Proceedings of the 11th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications* (2005), pp. 102–108.
[5] BAI, X., KUMAR, S., XUAN, D., YUN, Z., AND LAI, T. Deploying wireless sensors to achieve both coverage and connectivity. In *Proceedings of the 7th ACM international symposium on Mobile ad hoc networking and computing* (2006), pp. 131–142.
[6] BOLONI, L., AND TURGUT, D. Should i send now or send later? a decision-theoretic approach to transmission scheduling in sensor networks with mobile sinks. In *Wireless Communications and Mobile Computing* (2008), vol. 8.
[7] BRASS, P. Bounds on coverage and target detection capabilities for models of networks of mobile sensors. In *ACM Transactions on Sensor Networks* ($TOSN$) (2007), vol. 3, pp. 9–es.
[8] CHÉDIN, A., SAUNDERS, R., HOLLINGSWORTH, A., SCOTT, N., MATRICARDI, M., ETCHETO, J., CLERBAUX, C., ARMANTE, R., AND CREVOISIER, C. The feasibility of monitoring co2 from high-resolution infrared sounders. In *J. Geophys. Res* (2003), vol. 108, p. 4064.
[9] CHEN, D., DU, D., HU, X., LIN, G., WANG, L., AND XUE, G. Approximations for steiner trees with minimum number of steiner points. In *Journal of Global Optimization* (2000), vol. 18, pp. 17–33.

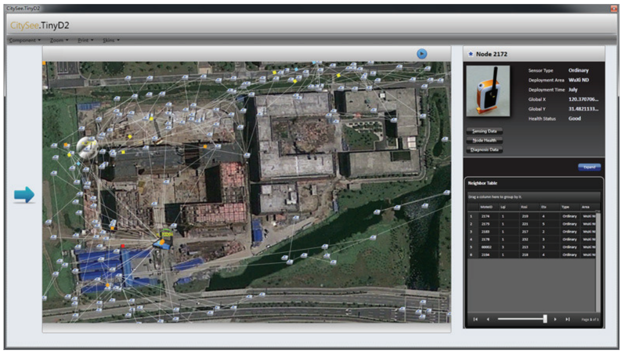


Fig. 12. A snapshot of CitySee.

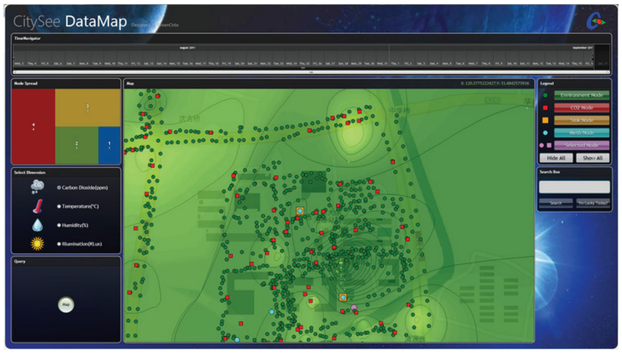


Fig. 13. Contour map of CO_2 concentration.

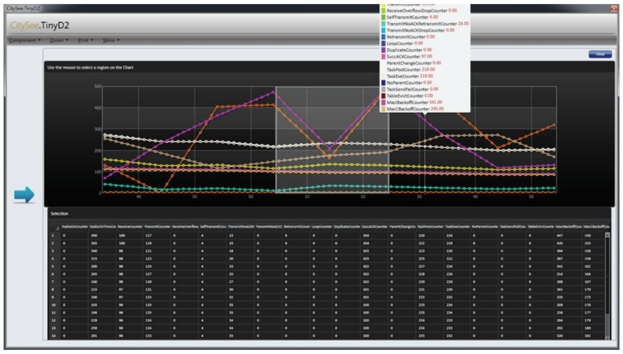


Fig. 14. Node's running status.

Fig. 15. System management interface.

[10] Cheng, X., Du, D., Wang, L., and Xu, B. Relay sensor placement in wireless sensor networks. In *Wireless Networks* (2008), no. 3, pp. 347–355.

[11] Churkina, G. Modeling the carbon cycle of urban systems. In *Ecological Modelling* (2008), vol. 216, pp. 107–113.

[12] Drake, D., and Hougardy, S. On approximation algorithms for the terminal steiner tree problem. In *Information Processing Letters* (2004), vol. 89, pp. 15–18.

[13] Fonseca, R., Gnawali, O., Jamieson, K., and Levis, P. Four-bit wireless link estimation. In *Proceedings of the Sixth Workshop on Hot Topics in Networks* $(HotNetsVI)$ (2007).

[14] Ganesan, D., Cristescu, R., and Beferull-Lozano, B. Power-efficient sensor placement and transmission structure for data gathering under distortion constraints. In *ACM Transactions on Sensor Networks* $(TOSN)$ (2006), vol. 2, pp. 155–181.

[15] Gnawali, O., Fonseca, R., Jamieson, K., Moss, D., and Levis, P. Collection tree protocol. In *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems* (2009), pp. 1–14.

[16] Halperin, E., Kortsarz, G., Krauthgamer, R., Srinivasan, A., and Wang, N. Integrality ratio for group steiner trees and directed steiner trees. In *Proceedings of the fourteenth annual ACM-SIAM symposium on Discrete algorithms* (2003), Society for Industrial and Applied Mathematics, pp. 275–284.

[17] Hao, B., Tang, H., and Xue, G. Fault-tolerant relay node placement in wireless sensor networks: formulation and approximation. In *IEEE HPSR* (2004), pp. 246–250.

[18] Kapoor, S., Maheshwari, S., and Mitchell, J. An efficient algorithm for euclidean shortest paths among polygonal obstacles in the plane. In *Discrete & Computational Geometry* (1997), no. 4, pp. 377–383.

[19] Lin, G., and Xue, G. Steiner tree problem with minimum number of steiner points and bounded edge-length. In *Information Processing Letters* (1999), vol. 69, pp. 53–57.

[20] Liu, K., Ma, Q., Zhao, X., and Liu, Y. Self-diagnosis for large scale wireless sensor networks. In *Proceedings of IEEE INFOCOM* (2011), pp. 1539–1547.

[21] Liu, Y., He, Y., Li, M., Wang, J., Liu, K., Mo, L., Dong, W., Yang, Z., Xi, M., Zhao, J., et al. Does wireless sensor network scale? a measurement study on greenorbs. In *Proceedings of IEEE INFOCOM* (2011), pp. 873–881.

[22] Liu, Y., Liu, K., and Li, M. Passive diagnosis for wireless sensor networks. In *IEEE/ACM Transactions on Networking* (TON) (2010), vol. 18, pp. 1132–1144.

[23] Lloyd, E., and Xue, G. Relay node placement in wireless sensor networks. In *IEEE Transactions on Computers* (2007), pp. 134–138.

[24] Maleki, M., and Pedram, M. Qom and lifetime-constrained random deployment of sensor networks for minimum energy consumption. In *Proceedings of the Fourth IEEE International Symposium on Information Processing in Sensor Networks* (2005), pp. 293–300.

[25] Mao, X., Miao, X., He, Y., Zhu, T., Wang, J., Dong, W., Li, X., and Liu, Y. Citysee: Urban co2 monitoring with sensors. In *Technical Report* (2012). Http://mypages.iit.edu/~xmao3/papers/CitySeeTechReport.pdf.

[26] Miao, X., Liu, K., He, Y., Liu, Y., and Papadias, D. Agnostic diagnosis: Discovering silent failures in wireless sensor networks. In *Proceedings of IEEE INFOCOM* (2011), pp. 1548–1556.

[27] Tang, J., Hao, B., and Sen, A. Relay node placement in large scale wireless sensor networks. In *Computer Communications* (2006), vol. 29, pp. 490–501.

[28] Tolle, G., and Culler, D. Design of an application-cooperative management system for wireless sensor networks. In *Proceedings of the Second IEEE European Workshop on Wireless Sensor Networks,* (2005), pp. 121–132.

[29] Wang, X., Fu, L., Tian, X., Bei, Y., Peng, Q., Gan, X., Yu, H., and Liu, J. Converge-cast: On the capacity and delay tradeoffs. In *IEEE Transactions on Mobile Computing* (2011), no. 99, pp. 1–1.

[30] Wang, Y., and Tseng, Y. Distributed deployment schemes for mobile wireless sensor networks to ensure multilevel coverage. In *IEEE Transactions on Parallel and Distributed Systems* (2008), vol. 19, pp. 1280–1294.