COPlanner: A Wireless Sensor Network Deployment Planning Architecture Using Unmanned Vehicles As Deployment Tools

Ramin Soleymani-Fard, Chia-Yen Shih, Marvin Baudewig and Pedro José Marrón
Networked Embedded Systems Group
Department of Computer and Cognitive Sciences
Universität Duisburg-Essen
Duisburg, Germany
{*ramin.soleymani-fard, chia-yen.shih, baudewig, pjmarron*}*@uni-due.de*

***Abstract*—Wireless sensor networks (WSNs) have been deployed for a variety of applications. As the scale of WSN deployments has largely increased and the application scenarios have become more complex, WSN deployment planning can save unnecessary expense on redundant hardware, software and human resources and thus makes the deployment more efficient. This paper presents an ongoing research on developing a deployment planning architecture, called *COPlanner*. Our goal is to provide deployment planning strategies for the static and mobile WSN applications to meet their requirements with respect to sensing coverage, network connectivity and data collection. Most importantly, for the inaccessible area, we consider using *Unmanned Aerial or Ground Vehicles* (UAVs or UGVs) as the deployment tools. Therefore, the planning architecture also covers the *Waypoint Planning* problem and offers scheduled routes for the autonomous vehicles to deploy sensor nodes and collect the data.**

***Keywords*-deployment planning; sensing coverage; connectivity; unmanned vehicles; waypoint planning; obstacle avoidance.**

I. INTRODUCTION

Wireless sensor networks (WSNs) have been deployed in physical environments (both indoor and outdoor areas) for a wide range of applications including habitant monitoring, disaster management, inventory control, etc. A WSN typically consists of a set of static or mobile sensor nodes, or so-called *Cooperating Objects* (COs) [1], which provide physical measurements and collaboratively carry out system operations to achieve the application objectives.

As the scale of current CO deployments becomes considerably large, e.g., thousands of sensor nodes and the application scenarios become more complex, effective and thorough pre-deployment planning is necessary and is the key to reduce the deployment cost. To provide a useful deployment plan, the developer must consider several aspects when designing an architecture for a WSN deployment tool. First, the planning architecture needs to allow the user to specify the application objectives, deployment requirement and the target area characteristics as well as the constraints for the deployment. Second, for most WSN applications with static nodes the main deployment requirements are centered around the issues of sensing coverage and network connectivity [2] [3]. Therefore, the architecture must offer a set of deployment planning strategies [4] [5] [6] and parameterization for optimizing the CO deployments in order to meet the application requirement. As for the application with mobile nodes [7], the architecture must provide the movement schedules as well as the trajectories for the mobile nodes to perform system operations in order to achieve required application performance. Third, the most important things is that the planning architecture has to include an evaluation unit for assessing the application performance based on the resulting CO deployments and to output the optimal deployment plan for the application.

We have been developing a WSN deployment planning architecture called *COPlanner*, which aims to cover the above aspects regarding static and mobile sensor node deployments. Moreover, COPlanner provides realistic deployment strategies by taking into account the physical obstacles. Furthermore, COPlanner considers using UAVs/UGVs as deployment tools for the inaccessible target areas. Therefore, the deployment plan generated by COPlanner also tackles the *Waypoint Planning* problem, which involves defining the optimal route for the deployment vehicles to visit a given set of waypoints as the deployment locations. COPlanner is developed in the scope of an European project, *PLANET* [8], in which the deployment planning tool is required to support the CO deployment using UAVs/UGVs for two applications: the *Wildlife Monitoring* in *Doñana Biological Reserve* (DBR) [9] and the *Automated Airfield* applications.

The remainder of the paper is structured as follows. We list the related work in Section II. Section III presents our architecture design of the COPlanner and it compositional components; Section IV describes the planning strategies we have developed for the static WSNs regarding sensor coverage and network connectivity, and for the mobile networks regarding waypoint planning. Finally, we describe our future work and conclude our work in Section V.

II. RELATED WORK

There have been lots of deployment strategies proposed to achieve sensing coverage and network connectivity. Many approaches are based on virtual forces, with which the nodes are either attracted or repulsed in order to achieve required coverage or connectivity. In [10], the distribution of sensing events was used to generate virtual forces, which shape the network topology to adapt the distribution of expected sensing events. Other grid-based approaches [5] [11] divided the target area into subareas and arranged the node locations in order to fulfill the deployment requirement. While these approaches aim to solve the deployment issues, they do not consider the impact of obstacles. All of our deployment strategies deal with the obstacle issue. Tan et al. [12] and Wang et al. [5] proposed algorithms for the sensor deployment to cover the target area, which can also include obstacles. Both approaches tried to minimize the number of nodes to be deployed. However, these approaches do not maintain k-connectivity.

While many deployment strategies are available, little work has been found on the WSN deployment planning environment. Li et al. described a planning platform [6] that guides the developer through three steps for deployment planning: *Pre-Placement*, *Simulation/Evaluation* and *Optimization*. The aim of the platform is to provide an integrated, general deployment planning environment. However, the deployment strategies for various applications were not addressed. The goal of COPlanner is to provide different deployment strategies to meet various application requirements. Therefore, more development efforts spent on the optimization deployment approaches for different network deployment configurations.

III. Architecture

The main functionality of COPlanner is to, given the user application input, create efficient deployment plans for various static and mobile WSNs applications. Figure 1 depicts an overview of the COPlanner architecture. In our design, the user can specify application-specific input and obtain the final deployment plan through a web-based application, so a web service client component is devised to handle the interaction with the web server. The user application input is maintained by the *User Application* (UA) component. Such input includes the target area description with the deployment constraints, application task description, planning configuration and sensing device configuration. Each piece of information is maintained by a corresponding subcomponent associated with the UA component.

The core component of COPlanner is the *Planning Manager*, which decides the type of the deployment plan to be generated based on the application input. Once the plan type is determined, the manager interacts with the *Planning Tool Manager* (PTM), which is implemented with a powerful scheme that allows flexibly extending the COPlanner's capability with newly developed deployment planning algorithms/strategies. PTM manages a set of *Planning Tools*, each of which implements an optimization deployment

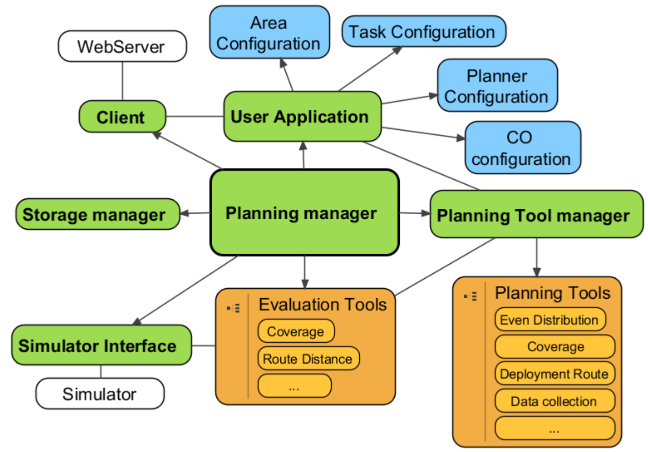

Figure 1. The Architecture of COPlanner

algorithm to meet the specific user requirements on the application performance, e.g., data delivery rate, latency and system lifetime. Section IV will detail these strategies.

The planning tool outputs one or a set of optimized pre-deployment plans created by the optimization algorithm. PTM then interacts with the *Evaluation Tools* component, which includes different performance metrics for evaluating the application performance based on the planned deployments. The evaluation of the pre-deployment plans is performed through the use of a WSN simulator. With the *Simulation Interface* component, the Evaluation Tool can specify the performance metrics and can request the simulator to simulate the application. From the application performance results with different pre-deployment plans, the Evaluation Tool decides on the *best* candidate as the output of the deployment plan. The final deployment plan as well as the performance evaluation is stored in the *Storage Manager* component. Note that the discussion on the simulator is not in the scope of the paper, and we focus our discussion on the architecture of COPlanner and the deployment strategies developed for generating optimized deployment plans.

IV. Planning Strategies

The design goal of COPlanner is to provide deployment strategies, which specify the node deployment locations for the WSN required by various applications. Our deployment strategies are classified into two kinds: (1) one for the static WSN deployments and (2) the other for the deployments involved in mobile vehicles, which can be further categorized into mobile sensor nodes and the deployment tools. These strategies are detailed below.

A. Static WSN Deployment Planning Strategies

Our deployment strategies for the static WSN deployment focus on the issues of sensing coverage and network connectivity with the minimum number of nodes. Moreover,

 ISBN: 978-1-61208-207-3

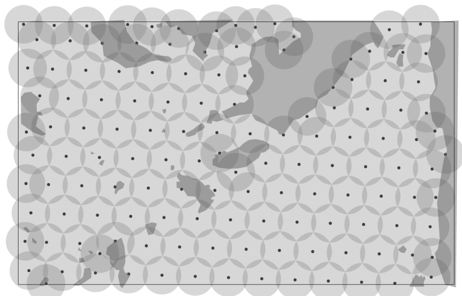

Figure 2. An example of generated *covered* deployment plan

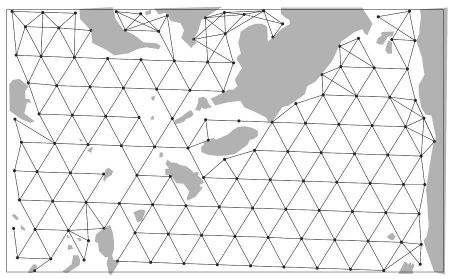

Figure 3. An examples of generated *connected* deployment plan

Figure 4. An example of generated route by WPP-O

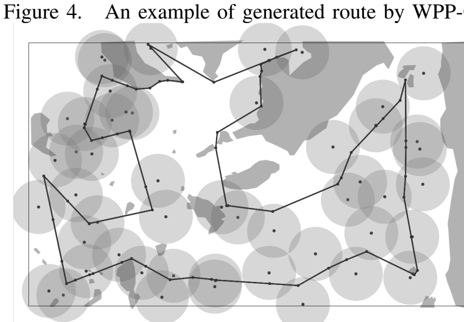

Figure 5. An example of generated route for data collection

different from the typical coverage and connectivity algorithms, our approach considers the existence of obstacles in the target area. In COPlanner, the obstacles are modeled as polygons, in which no deployment locations can be specified. We use the *Unit Disc Model* (UDM) for the sensing and communication coverage. We assume that two nodes A and B are *connected* if their Euclidean distance $|AB| \leq r_c$, where r_c is the communication range. However, A and B are *disconnected* if the segment $\overline{AB}$ intersects with any obstacle polygon. Note that our approaches, except for the CCP approach (described below), are not limited by the use of UDM. To simply the explanation without loss of generality, we use the UDM to show the resulting deployment plans.

We first developed the *Coverage and Connectivity Planning* (CCP) strategy, which generates a deployment plan with node deployment locations such that the every point (except for the ones in the obstacles) in the target area is covered, while the nodes stay connected. To achieve the minimum node number, we used the approach proposed in [4], in which the author analyzed the patterns of node locations in order to maintain *full coverage* and *k-connectivity* ($k \leq 6$) with the minimum number of nodes. However, when considering obstacles, the regularity of the pattern cannot be totally applied. We modified the approach as follows. Initially, our CCP approach uses the *hill climbing* technique to search for the pattern configuration, which identifies the starting node location and the angle for applying the pattern across the target area. In the case where the deployment locations fall into the obstacle ranges, the deployment *holes* will occur. To cope with this problem, CCP uses a heuristic algorithm to fill up the holes with the minimum number of nodes. Furthermore, if k-connectivity is required, additional nodes will be included to match the requirement. Figure 2 and Figure 3 illustrate the examples of resulting deployment plans for coverage and connectivity with the coverage radius $r_s = 50m$ and $r_c = 120m$, respectively.

Another deployment strategy that is currently in development involves evenly distributing the nodes across the target area. The motivation for this approach is the applications that requires sampling at different locations in the target area, e.g., water samples from different locations of a flooded area for pollution detection. Connectivity is not main requirement for such application and the data can be collected, e.g., using mobile elements. Wang describes in [13] a vector-based approach called Minimax Algorithm, which creates a uniformly distributed network topology. The main idea is to use a *Voronoi* diagram and the *minimax* points within the Voronoi cells to organize the originally randomly deployed nodes until the Voronoi cells have approximately the same size. The next step of our work is to adjust the algorithm so that it considers obstacles in the target area.

B. Mobile WSN Deployment Planning Strategies

In addition to the static WSN deployments, COPlanner also covers deployment planning for the mobile WSN applications. Moreover, COPlanner considers using the moving vehicles as the deployment tools. In this case, the deployment plan contains the locations to be visited and the movement schedules for the deployment vehicles and the

 ISBN: 978-1-61208-207-3

mobile nodes. The former can be generalized to the *Waypoint Planning* problem, in which given a set of waypoints, the algorithm outputs the optimal route to visit all the waypoints.

We developed a solution algorithm for the extended Waypoint Planning problem with the obstacles (WPP-O) and also modeled the obstacles as polygons in this case. In addition to the waypoints, our approach first identifies each vertex of the polygons and includes them in the graph G. The algorithm then finds the shortest paths between each pair of nodes in G using our modified Dijkstra algorithm. In the next step, our approach randomly selects a route that covers all waypoints, and tries to find the optimal route using the *Traveling Salesman Problem (TSP) with Simulated Annealing* technique, which assigns the distance as a cost value to each link. The goal is to create a route with the minimum cost value. In each iteration, a pair of links are switched to form a new route. The switch is accepted if the resulting cost is lower or with a certain probability if the cost is higher. When the iterations terminates, our algorithm outputs an optimized route that covers all waypoints while the obstacles are avoided. Figure 4 shows an example of the optimal route generated by our WPP-O algorithm.

Additionally, we are currently developing a deployment strategies for data collection with mobile nodes. The problem of data collection using mobile nodes is defined as: given a set of node locations and the communication range, the algorithm outputs a route for the mobile node to pass through the communication range of every node in order to collects all sensory data. The objective is also to minimize the distance of the route. Figure 5 illustrates an example of resulting route. More improvement on this approach still needs to be made in order to create the optimal route.

V. Conclusion and Future work

In this paper, we presented our ongoing work on a flexible deployment planning architecture (COPlanner) that can be easily extended to include different deployment strategies for various WSN applications using the static or mobile sensor nodes. Particularly, COPlanner provides deployment plans that involve in autonomous vehicles as the deployment tools. We believe that the development of COPlanner can provide a useful WSN deployment planning architecture to ease the WSN deployment task and to reduce the deployment cost.

In the future, we plan to perform thorough experiments to evaluate the developed algorithms, and further to use these techniques on the real deployment in the project. Moreover, we will consider more deployment approaches for more complex applications that required deployments of hybrid WSNs with both static and mobile nodes.

Acknowledgment

This work has been partially supported by EU PLANET, *PLAtform for the deployment and operation of heterogeneous NETworked cooperating objects*, and CONET, *the Cooperating Objects Network of Excellence* (PLANET: FP7-2009-5-257649, CONET: FP7-2007-2-224053).

References

[1] P. J. Marrn, S. Karnouskos, D. Minder, and the CONET consortium, *Research Roadmap on Cooperating Objects*. Office for Official Publications of the European Communities, 2009.

[2] C.-F. Huang and Y.-C. Tseng, “The coverage problem in a wireless sensor network,” *Mob. Netw. Appl.*, vol. 10, no. 4, pp. 519–528, Aug. 2005.

[3] A. Konstantinidis and K. Yang, “Multi-objective k-connected deployment and power assignment in wsns using a problem-specific constrained evolutionary algorithm based on decomposition,” *Comput. Commun.*, vol. 34, pp. 83–98, Jan. 2011.

[4] X. Bai, D. Xuan, Z. Yun, T. H. Lai, and W. Jia, “Complete optimal deployment patterns for full-coverage and k-connectivity (k<=6) wireless sensor networks,” in *Proceedings of the 9th ACM international symposium on Mobile ad hoc networking and computing*, 2008, pp. 401–410.

[5] Y.-C. Wang, C.-C. Hu, and Y.-C. Tseng, “Efficient deployment algorithms for ensuring coverage and connectivity of wireless sensor networks,” in *Proceedings of the First International Conference on Wireless Internet*, 2005, pp. 114–121.

[6] J. Li, Y. Bai, H. Ji, J. Ma, Y. Tian, and D. Qian, “Power: Planning and deployment platform for wireless sensor networks,” in *Grid and Cooperative Computing Workshops, GCCW ’06. Fifth International Conference on*, 2006, pp. 432–436.

[7] Y. Wang and C.-H. Wu, “Robot-assisted sensor network deployment and data collection,” in *Computational Intelligence in Robotics and Automation, 2007. CIRA 2007. International Symposium on*, june 2007, pp. 467 –472.

[8] The PLANET project, PLAtform for the deployment and operation of heterogeneous NETworked cooperating objects. [Online]. Available: www.planet-ict.eu

[9] Doñana Biological Reserve. [Online]. Available: http://www.ebd.csic.es/website1

[10] C. Koutsougeras, Y. Liu, and R. Zheng, “Event-driven sensor deployment using self-organizing maps,” *Int. J. Sen. Netw.*, vol. 3, pp. 142–151, May 2008, printed.

[11] J.-J. Chang, P.-C. Hsiu, and T.-W. Kuo, “Search-oriented deployment strategies for wireless sensor networks,” in *Object and Component-Oriented Real-Time Distributed Computing. 10th IEEE International Symposium on*, 2007, pp. 164–171.

[12] G. Tan, S. A. Jarvis, and A.-M. Kermarrec, “Connectivity-guaranteed and obstacle-adaptive deployment schemes for mobile sensor networks,” *Distributed Computing Systems, 2008. ICDCS ’08. The 28th International Conference on*, vol. 8, no. 6, pp. 836–848, 2008.

[13] G. Wang, G. Cao, and T. La Porta, “Movement-assisted sensor deployment,” in *INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies*, vol. 4, 2004, pp. 2469–2479.