MobiCom Poster Abstract: Asymmetric Sensing by Decoupling Scheduling from Switching

Yu Gu and Tian He
{yugu,tianhe}@cs.umn.edu
University of Minnesota, Twin Cities

I. Introduction

As a key approach to achieve energy efficiency in sensor networks, sensing coverage has been studied extensively. Researchers have designed many coverage protocols to provide various kinds of service guarantees within the network. While the state-of-the-art is encouraging, we believe there are some aspects that need further investigation. First, currently different sensing coverage algorithms focus on different service guarantees. Any single design is not general enough to meet a wide range of sensing requirements under different operating scenarios. Second, lifetime extension in most algorithms is essentially achieved through coordination among neighboring nodes. The local node density, therefore, imposes a theoretical upper bound on the system lifetime.

To address these two issues simultaneously, in this abstract, we propose a Unified Sensing Coverage Architecture, called uSense, which features three novel ideas: *Asymmetric Architecture*, *Generic Switching* and *Global Scheduling*. We propose asymmetric architecture based on the conceptual separation of switching from scheduling. Such asymmetric architecture enables us to design sophisticated coverage algorithms in an unconstrained design space, represent such intelligence with a lightweight algorithm implemented in sensor nodes and achieve a fast and efficient change of coverage algorithms by disseminating only several parameters. To our best knowledge, the proposed *generic switching* is the first generic and lightweight switching algorithm for resource-limited sensor nodes. As an instance, we propose a two-level global coverage algorithm, called uScan. Our initial evaluation results indicate that uSense is a promising architecture to support flexible and efficient coverage in sensor networks.

II. uSense Architecture

The uSense design consists of two parts as shown in Figure 1. The switching algorithm is implemented in the sensor nodes, which takes only two scheduling parameters as input: *working schedule bits: S* - an infinite binary string in which 1 denotes the active state and 0 denotes the inactive state, and *switching rate: R* - defines the rate of toggling between states. The computational entity is responsible for generating the scheduling parameters. It takes the schedules decided by sensing coverage algorithms and translates them into the parameters used by the switching algorithm. In order to efficiently disseminate these two parameters to the sensor nodes, we propose to express the *schedule bits*, which is usually periodic and following a certain pattern, in the form of regular expression. To further reduce communication costs, we use a parameterized schedule, which is a binary regular expression that changes with the node's location and enables us to disseminate the schedules through one limited flooding instead of unicasting to every single node.

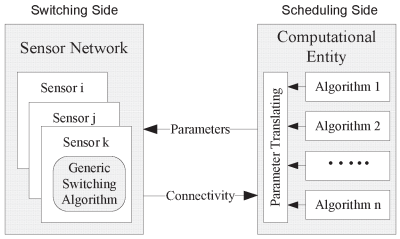

Fig. 1: Overview of uSense Architecture

III. Global Scheduling Algorithms

Conceptually, uSense can support many existing coverage algorithms. Due to its asymmetric architecture, it is especially friendly to the global scheduling algorithms. In this section, we describe a two-level global scheduling algorithm called uScan, which is one of sensing coverage algorithms in Figure 1 that are supported by the uSense architecture.

III.A. Level I: Tile Scheduling

In uScan, we partition an area under surveillance into some small regions of the same shape, a process called tessellation. These small regions are called *tiles*, which can be regular triangles, rectangles or regular hexagons in a 2-D space. The size of tiles is set to be smaller than the minimum target size, so that a target

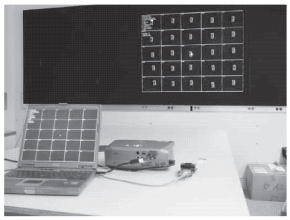

Fig. 2: uSense System Setup

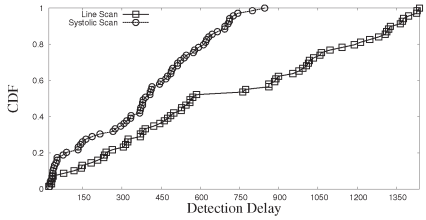

Fig. 3: Detection Delay

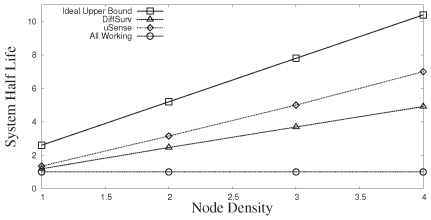

Fig. 4: Half Life vs. Node Densities

is detected as long as a portion of a tile is covered. As a reminder, nodes within a sensor network only support a generic switching algorithm, which has neither the concept of tiles nor the partition information of the tiles. All the complex logic resides in the computational entity. Here, we introduce two methods for the tile-level scheduling: 1) **Line Scan**, which schedules node only to cover a column/row of tiles in a certain interval of time during one round of scan. The covered columns/rows are increasing or decreasing consecutively; 2) **Systolic Scan** emulates the cardiac cycles of a beating heart. Over the area under surveillance, we sense the tiles from the inner layer to the outer layer.

Both line scan and systolic scan specify only the set of tiles need to be activated (covered) at a given point of time. The task of covering each tile set is accomplished by the second-level node scheduling, which is described in the next section.

III.B. Level II: Node Scheduling

The main idea of our node scheduling algorithm is to find the optimal set of nodes which could cover all the tiles that need to be active at time interval i. Before node scheduling, we first map physical node coverage into the coverage bipartite graph according to the coverage relationship. Then we divide node scheduling into two steps. First, for a tile set TS_i active at time i, we keep identifying 1-cover set with minimal number of nodes, until the size of 1-cover set is above a certain threshold. Secondly, we create schedules for nodes such that each identified 1-cover set provides coverage to TS_i in a round-robin fashion.

The generic Minimum Set Cover problem has been proven NP-Hard [1]. Fortunately, we find line scan coverage is a special case of the generic set cover problem, because a node only need to cover a continuous segment of tiles. By mapping the coverage bipartite graph into a directed acyclic graph, we could reduce the tile set cover problem to the problem of finding out the shortest paths between the start and end of a tile set TS_i, with a runtime of $O(|V|) + O(|E|)$.

We note that the proposed polynomial algorithm does not apply to generic tile scheduling, where a tile set does not form a continuous curve or where a node can cover multiple segments of a tile set simultaneously. In these cases, we adopt a greedy set-cover method by choosing the node that covers the most number of tiles first.

In order to support line scan or systolic scan in a 2-D space, we need to identify cover sets for the whole area(not just for a single tile set). Thus a node may need to cover multiple tile sets TS_i. To effectively handle the cases where we have to select cover sets for multiple TS, we designed an algorithm that each node maintains a counter SC which records how many times it has been selected into a unique cover sets. While selecting cover sets for each tile set TS, instead of solely consider the number of nodes in the cover sets, the algorithm calculates the minimum cover set among the nodes whose SC counter values are as small as possible.

After obtaining cover sets for every tile set TS_i, we build the final schedule of node according to all the cover sets it's belonged to, which can be executed directly by our generic switching algorithm.

IV. Implementation and Evaluation

We have implemented a complete version of uSense on Berkeley TinyOS/Mote platform, using 30 MicaZ motes as shown in Figure 2. The results showed that uSense is a lightweight and efficient architecture. To reveal the system performance at scale, we have conducted some initial large scale simulations with 10,000-node. Under full coverage mode, we demonstrated that our global scheduling algorithms provide significant energy savings over previous protocols such as DiffSurv [2] under metrics such as Half-life, Coverage Overtime and Node Energy Consumption.

References

[1] V. T. Paschos. A Survey of Approximately Optimal Solutions to Some Covering and Packing Problems. In *ACM Computing Surveys*, June 1997.

[2] T. Yan, T. He, and J. A. Stankovic. Differentiated Surveillance Service for Sensor Networks. In *SenSys'03*, November 2003.