Demo: Distributed MaxRS in Wireless Sensor Networks*

Muhammed Mas-ud Hussain, Panitan Wongse-ammat, and Goce Trajcevski
Department of Electrical Engineering and Computer Science
Northwestern University
Evanston, IL 60208
{mmh683, pwa732, goce}@eecs.northwestern.edu

ABSTRACT

This work presents a distributed implementation for processing *Maximizing Range Sum* (MaxRS) query in Wireless Sensor Networks (WSN). MaxRS query is useful in many spatially-distributed event monitoring and target tracking applications. Given the location and current readings of the nodes, and a rectangle R, MaxRS finds a location of R that maximizes the sum of the readings of all the nodes covered by R. Our system performs MaxRS query in a user-specified time-interval γ and using the result obtained, attempts to maintain a certain degree of energy conservation in the WSN, based on a user-defined threshold δ. Since centralized processing of the raw readings and subsequently determining the MaxRS may incur significant communication overheads, we developed a distributed algorithm to compute MaxRS. We implemented our system in a heterogeneous WSN consisting of TelosB and SunSPOT motes, and illustrate the end-user tools: GUI for specifying required parameters, and real-time visualization of MaxRS solutions and estimated network energy consumption.

1. INTRODUCTION

Wireless Sensor Networks (WSN) represent a paradigm with broad range of practical applications, including but not limited to global-scale environmental monitoring, smart buildings and cities, medicine, agriculture, safety and hazard detection, etc. [1]. As sensor nodes are usually powered by batteries and often severely limited for recharging or replacing them, reducing the energy consumption is an ever-important topic in WSN, enabling an extension of overall network's operational lifetime [2]. While periodic sampling and transmission to a dedicated sink may be well-suited for certain applications, they may incur significant overheads in others—especially in event-based monitoring. Various works have tackled routing issues, as well as epoch-based synchronization for query processing, aggregation, and in-network algorithms to minimize communication overheads [3].

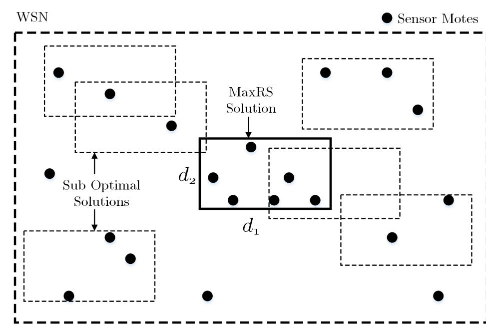

Figure 1: An example of MaxRS query in WSN.

The specific problem that we address can be described as follows. We are given a set O of *weighted objects* and a rectangle R of a specific dimension (i.e., $d_1 \times d_2$), and the *Maximizing Range Sum* (MaxRS) retrieves a location of R within the given space that maximizes the sum of the weights of all the objects covered by R. In the case of WSNs, we can think of the set of sensor nodes as the set of weighted objects. The weights of the individual sensor nodes may be set based on specific application needs, e.g., information gain (tracking an object), mote readings (event monitoring), uniform (counting), etc. An instance of the MaxRS in WSN is shown in Fig. 1, where the weights of all the sensor nodes are uniform.

MaxRS query can be used effectively for energy conservation purposes in applications such as identifying and monitoring the region with the highest density of tracked objects (e.g., gazelles); tracking the portion of the city with heaviest traffic; detecting most critical sources of forest fires (locating hotspots) and tracing their expansion routes; etc.

Our goal is to provide efficient mechanisms for: (1) Processing MaxRS query in WSN; and (2) Reducing energy consumption while maintaining its answer under dynamically changing values of the monitored phenomena. Towards this, we have developed an in-network algorithm to compute the MaxRS query, conserving the energy spent by awaking only the nodes that are within the MaxRS solution, and keeping all others asleep. We also limit the total network energy consumed in a uniformly or randomly deployed WSN to a certain degree based on a user-specified threshold by adjusting the shape of the query rectangle R accordingly. This demo paper exhibits the system that implements these

*Research supported in part by NSF grants CNS-0910952 and III 1213038, and ONR grant N00014-14-1-0215.

SenSys '15, November 1–4, 2015, Seoul, Republic of Korea.
ACM 978-1-4503-3631-4/15/11.
DOI: http://dx.doi.org/10.1145/2809695.2817863.

techniques on a network of heterogeneous motes (TelosB and SunSPOT), along with the user-interface to specify required parameters and modules for visualizing the event monitored, MaxRS results, and amount of energy consumed.

2. SYSTEM DESIGN

Two parameters in our system control how the energy efficiency is enforced:

• γ — corresponds to the time-interval of MaxRS query processing. If MaxRS is computed at time t using the current set of weights of the sensor nodes, only the nodes within the solution rectangle are kept awake between time t and $t+\gamma$. At time $t+\gamma$, MaxRS is re-computed with updated weights, and state (awake, or idle) of the nodes are changed accordingly.

• δ — denotes the target average network energy consumption. The shape of the MaxRS query rectangle is adjusted based on δ. We assume that the overall energy spent at any time is directly proportional to the number of active sensor nodes in the network. As a simple model, we assume that the whole WSN area is a $l_1 \times l_2$ meters^2 rectangle, where n sensor nodes are deployed uniformly. If k nodes total to δ energy at a time instant, then the shape of the query rectangle R would be $d_1 \times d_2$, where $d_i = \frac{k \times l_i}{n}$ $(i \in \{1, 2\})$.

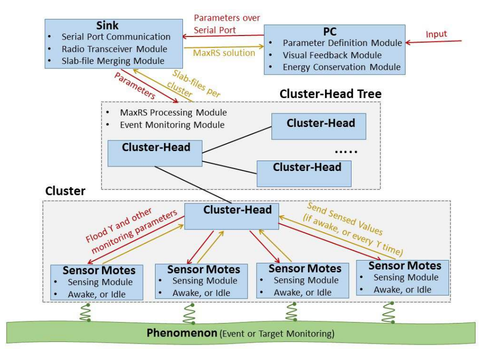

Figure 2: System Architecture.

The distributed solution for the MaxRS problem in spatial databases in [4] is the foundation of our in-network algorithm. Instead of dividing the space into m vertical slabs, we divide the space into a hierarchy of $m \times n$ grid-shaped clusters (similar to [3]), each of which has a selected cluster-head. The system architecture shown in Fig. 2 consists of the following main components:

(1) At the highest declarative level, we have the GUI that serves two main purposes: (a) Enables the users to select desired values for the required parameters; and (b) Provides a display for visualizing the boundaries of the current MaxRS solution, showing the nodes that are currently awake; the status of the monitored event (e.g., mote readings); and energy consumption in the network.

(2) The sink, connected through the serial port of the laptop: (a) Disseminates the parameters to the cluster-heads; and (b) Collects the data from the cluster-heads and performs the slab-file merging procedure described in [4].

(3) The cluster-heads form a tree-based hierarchy towards the sink. Each cluster-head calculates the slab-file information for the local cluster and coordinates the information with one of the neighboring cluster-heads to perform

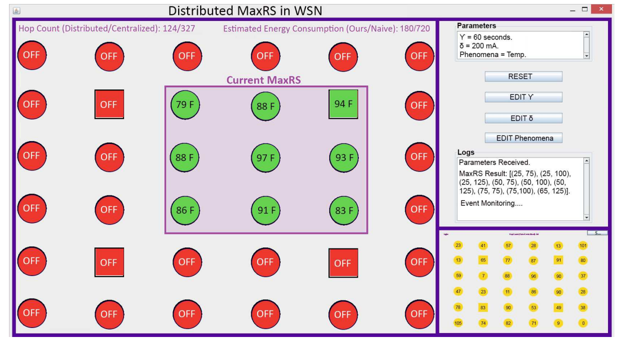

Figure 3: Graphical User Interface.

in-network aggregation.

(4) At the lowest level, the individual motes in each cluster conduct two simple tasks: (a) They receive related parameters from the local cluster-head; and (b) When awake, they report their readings to the local cluster-head.

3. DEMO SPECIFICATIONS

Our demo setup will consist of an actual WSN composed of 5 SunSPOT and 32 TelosB motes, connected via the sink to a USB port of a laptop running the Java-based front-end application. The motes will be distributed uniformly, and the WSN will be organized into 4 clusters having 8 TelosB motes and 1 SunPOT (cluster-head) each. The remaining SunSPOT will be used as the sink. We will use two different phenomena for event monitoring and sensing: temperature and light. The demonstration will have two distinct parts:

P1: The first part will have the following two main phases:

Phase-1: Specification of the required parameters for energy conservation and event monitoring in the GUI, illustrating the means to control the value of: (a) γ; (b) δ; and (c) Phenomena (light, or temperature).

Phase-2: Demonstration of the real-time execution in the provided WSN based on the user-defined parameters obtained in *Phase-1*. We will display (cf. Fig. 3): (a) Current MaxRS solution; (b) Currently active motes and their readings; and (c) Effect of the values of γ and δ on (a) and (b).

P2: The second part of the demo will compare the estimated energy consumption: (a) Using MaxRS results vs. keeping all nodes awake; and (b) Centralized vs. distributed processing of MaxRS query (via total message hop count).

4. REFERENCES

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks: a survey. *Comput. Netw.*, 38(4):393–422, Mar. 2002.

[2] G. Anastasi, M. Conti, M. Di Francesco, and A. Passarella. Energy conservation in wireless sensor networks: A survey. *Ad hoc networks*, 7(3):537–568, 2009.

[3] B. Avci, B. Zhang, M. M. Hussain, and G. Trajcevski. Evolving shapes in wireless sensor networks. In *Proceedings of the 12th ACM Conference on Embedded Network Sensor Systems*, pages 320–321. ACM, 2014.

[4] D. W. Choi, C. W. Chung, and Y. Tao. Maximizing range sum in external memory. *ACM Trans. Database Syst.*, 39(3):21:1–21:44, Oct. 2014.