Demo Abstract: Evolving Shapes in Wireless Sensor Networks

Besim Avci, Bing Zhang, Muhammed Mas-ud Hussain, and Goce Trajcevski*
Dept. of EECS
Northwestern University
Evanston, IL, USA

{bna728, bzv686, mmh683, goce}@eecs.northwestern.edu

Abstract

We present an implementation of a system for managing evolving shapes in Wireless Sensor Networks (WSN). A shape is a contiguous region in which the measurements of the sensors detect values above a given threshold. Our system, in its current version, solves two important problems: (1) Detecting and tracking the changes of boundaries; (2) Detecting an occurrence of *within distance* predicate for two (or more) shapes. A centralized approach (transmitting raw measurements to a dedicated sink) incurs communication overhead, so we developed distributed algorithms for managing the predicates related to evolving shapes. This demo will present the implementation of our solutions in a heterogeneous WSN consisting of TelosB and SunSPOT motes. It will also illustrate the end-user tools: interface for specifying the parameters of the predicates, along with real-time visualization of their evaluation.

1 Introduction

Wireless Sensor Networks (WSN) are one of the most suitable technological tools for tasks in plethora of domains, ranging from environmental and habitat monitoring, through (structural) safety and hazard detection, to various medical applications [5]. While periodic sampling and transmission to a dedicated sink may be well-suited for settings requiring a complete coverage of a given field, they may incur substantial overheads – and several works have proposed routing, epoch-based synchronization, aggregation and in-network processing mechanisms to minimize such overheads (e.g., [4]). Complementary to this, in many applications the task of interest amounts to detecting events – i.e., occurrence of "something of interest". Contrary to the continuous monitoring, in event-based applications the approaches of transmitting raw data to the sink generate even larger unnecessary overhead. Hence, it is a paramount to have efficient distributed in-network algorithms for detecting such events.

The context of our main objectives was phenomena characterized by spatial extents – e.g., boundaries of a region in which the temperature readings exceed 75°C; etc... Several research attempts addressed the problem of capturing the shapes characterizing the regions in which sensors' readings satisfy certain properties – e.g., isocontour map construction [3] or boundary detection [2]. Our specific goal was to provide efficient mechanisms for: (1) Tracking the evolution of shapes which change over time; and (2) Detecting a co-occurrence of two (or more) such evolving shapes, within certain proximity of each other. Co-occurrence detection is of interest in many applications – e.g., two forest fire regions coming to each other within a certain distance is crucial for evacuation purposes and adapting the arrangement of the fire-trucks. Towards this, in [1] we introduced distributed in-network methodologies for detection and tracking of spatial shapes and their co-occurrences. This demo paper presents the system which implements the techniques proposed in [1] on a network of heterogeneous motes (TelosB and SunSPOT), along with the GUI for users to specify the (parameters of the) desired predicates and modules for visualizing the detection of the predicates in the sink.

2 System Overview

We have two types of predicates which specify the properties of the shapes to be detected/tracked and their co-occurrence:

• $E_S(\gamma, A, t)$ – this predicate corresponds to an occurrence of an event denoting a detection of a connected spatial shape (S) such that: the smallest value read by every sensor inside S is at least γ; the area of S is $\geq A$; and the time of detection of E_S is t.

• $E_{CO}(E_{S1}, E_{S2}, d, \delta)$ – this composite event holds iff $E_{S1}(\gamma_1, A_1, t_1)$ and $E_{S2}(\gamma_2, A_2, t_2)$ are detected in a WSN and in addition: $|t_1 - t_2| \leq \delta$; the Euclidean distance between the two closest points of the boundaries of S_1 and S_2 is $\leq d$.

The software architecture of our system, illustrated in Figure 1, is organized in the following main categories:

(1) At the highest declarative level is the Graphical User Interface (GUI) the main purpose of which is twofold: (a) It provides means for the users to select the desired predicate

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

SenSys'14, November 3–5, 2014, Memphis, TN, USA.
ACM 978-1-4503-3143-2/14/11.
http://dx.doi.org/10.1145/2668332.2668376

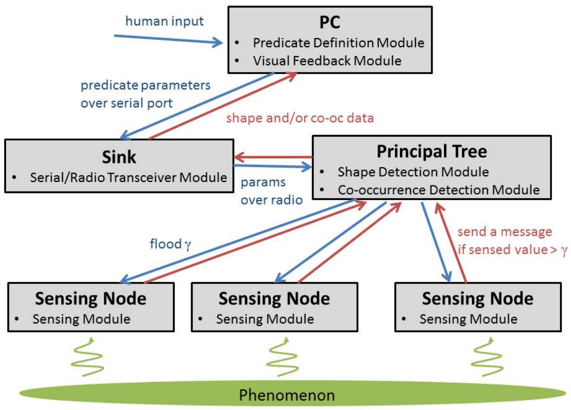

Figure 1. Main Modules

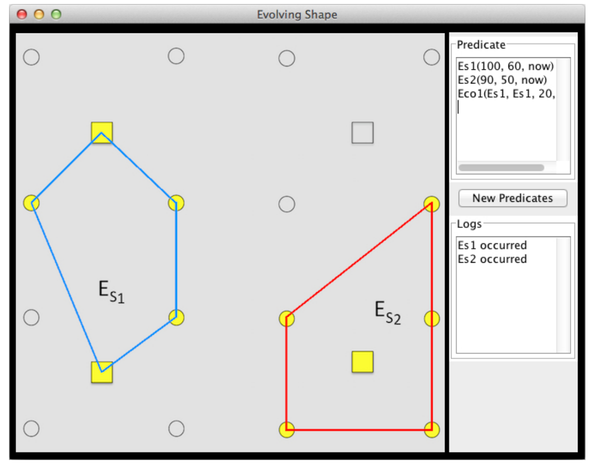

Figure 2. GUI – Visualization Instance

i.e., to describe the parameters of the events of interests – γ, A for E_S type, or the corresponding values for E_{S1} and E_{S2} along with d and δ for the E_{CO} type; (b) It has a display for visualizing the boundaries of the polygonal approximation of the shapes satisfying the corresponding predicates. In the case of E_{CO}, the line segment illustrating the shortest distance is also displayed. This front-end interface is a Java application residing on a laptop. In the case of "outliers" (nodes sensing values $\geq \gamma$ but with coverage area $< A$) the display will "blink" such nodes.
(2) The sink, connected through the serial port: (a) disseminates the predicates' parameters to the cluster-heads; (b) collects the data from the cluster-heads and connects it with the display module upon detection of the events. The motes are organized in a hierarchy of clusters, each having a selected principal (cluster-head) and the principals form a tree-based hierarchy towards the sink. Each principal is in charge of coordinating the values of the (readings of the) motes in its cluster, for the purpose of detecting a potential occurrence of the desired predicate. In the case that the "local" area of detection is $< A$, the principal will collaborate with its siblings to check if merging could result in an area $\geq A$. Also, messages are transmitted in a multi-hop manner (even though motes are close enough for direct communication for demo).
(3) The individual motes in each cluster performs two simple tasks: (a) they receive the threshold values from the local cluster-head; (b) they report their readings (and the timestamp) anytime the underlying phenomenon of interest has value which is sensed to be $\geq \gamma$.

3 Demo Specifications

The setup of our demo will consist of a laptop running the Java-based front-end application, connected to the sink of an actual WSN consisting of 5 SunSPOT motes and 16 TelosB motes. The motes will have a uniform geographic distribution, and will be organized in 4 main clusters consisting of 4 TelosB motes and 1 SunSPOT each, where the SunSPOT will serve the role of the principal for a given cluster. The last SunSPOT mote, having the role of a sink, will be connected to the serial port of the laptop. We will consider two different phenomena for specifying the threshold-values for the sensed measurements: temperature and light. The step-by-step demonstration will have two distinct parts.
P1: The first part will have the following three main phases:
Ph1: Illustrate the specification of the parameters of the desired events in the GUI. This phase will show both: (a) Definition of simple events (for each phenomenon); and (b) Definition of the composite (co-occurrence) event.
Ph2: In this phase, for each of the specifications in *Ph1(a)* and *Ph1(b)* we will demonstrate the actual execution in the provided WSN. Towards that, we will show three different scenarios: (a) Outliers – notification of certain readings exceeding the desired threshold(s), but with area covering $< A$; (b) Detection of satisfiability of instances of E_S as specified in *Ph1(a)* (cf. Figure 2); and (c) Detection of the satisfiability of an instance of E_{CO} as specified in *Ph1(b)*.
P2: The second part of the demo will show the total message hop count for different execution-scenarios (cf. *Ph2*) in two main settings: (a) centralized (which is, every single reading is transmitted to the sink, and the predicates are evaluated at the host laptop); and (b) distributed – as we described in Section 2.

4 References

[1] B. Avci, G. Trajcevski, and P. Scheuermann. Managing evolving shapes in sensor networks. In *SSDBM*, 2014.

[2] S. Duttagupta, K. Ramamritham, and P. Kulkarni. Tracking dynamic boundaries using sensor network. *IEEE Trans. PDP*, 22(10), Oct 2011.

[3] S. Gandhi, J. Hershberger, and S. Suri. Approximate isocontours and spatial summaries for sensor networks. In *IPSN*.

[4] S. Madden, M. Franklin, J. Hellerstein, and W. Hong. Tinydb: An acquisitional query processing system for sensor networks. *ACM TODS*, 30(1), 2005.

[5] R. Szewczyk, A. M. Mainwaring, J. Polastre, J. Anderson, and D. E. Culler. An analysis of a large scale habitat monitoring application. In *SenSys*, pages 214–226, 2004.