Pervasive and Mobile Computing 5 (2009) 135–147

Contents lists available at ScienceDirect

Pervasive and Mobile Computing

journal homepage: www.elsevier.com/locate/pmc

JAID: An algorithm for data fusion and jamming avoidance on distributed sensor networks

Aristides Mpitziopoulos [a,*], Damianos Gavalas [a], Charalampos Konstantopoulos [b], Grammati Pantziou [c]

[a] Department of Cultural Technology and Communication, University of the Aegean, Lesvos, Greece

[b] Research Academic Computer Technology Institute, Patras, Greece

[c] Department of Informatics, Technological Educational Institution of Athens, Athens, Greece

ARTICLE INFO

Article history:
Received 8 October 2007
Received in revised form 6 May 2008
Accepted 15 June 2008
Available online 24 June 2008

Keywords:
Wireless sensor networks
Mobile agents
Jamming avoidance
Routing
Data fusion
Itineraries

ABSTRACT

Mobile Agent (MA) technology has been recently proposed in Wireless Sensor Networks (WSNs) literature to answer the scalability problem of client/server model in data fusion applications. In this paper, we describe the critical role MAs can play in the field of security and robustness of a WSN in addition to data fusion. The design objective of our Jamming Avoidance Itinerary Design (JAID) algorithm is twofold: (a) to calculate near-optimal routes for MAs that incrementally fuse the data as they visit the nodes; (b) in the face of jamming attacks against the WSN, to modify the itineraries of the MAs to bypass the jammed area(s) while not disrupting the efficient data dissemination from working sensors. If the number of jammed nodes is small, JAID only modifies the pre-jamming scheduled itineraries to increase the algorithm's promptness. Otherwise, JAID re-constructs the agent itineraries excluding the jammed area(s). Another important feature of JAID is the suppression of data taken from sensors when the associated successive readings do not vary significantly. Data suppression also occurs when sensors' readings are identical to those of their neighboring sensors. Simulation results confirm that JAID enables retrieval of information from the working sensors of partially jammed WSNs and verifies its performance gain over alternative approaches in data fusion tasks.

1. Introduction

Mobile Agent (MA) technology represents a relatively recent trend in distributed computing, which answers the flexibility and scalability problems of centralized schemes. The term MA [18] refers to an autonomous program with the ability to move from host to host and act on behalf of users towards the completion of an assigned task. A MA may be defined as an entity that comprises a data space to carry collected values and an itinerary which can either be fixed or dynamically determined based on the current network status and the application logic (or execution code) [19]. Lange and Oshima listed seven good reasons to use MAs [12]: reducing network load, overcoming network latency, robust and fault-tolerant performance, etc. The MAs-based computing model enables moving the code (processing) to the data rather than transferring raw data to the processing element.

Although the role of MAs in distributed computing is still being debated mainly due to security concerns [8], several applications have shown clear evidence of benefiting from the use of MAs [15], including e-commerce and m-commerce

* Corresponding author. Tel.: +30 22510 36643.
E-mail addresses: crmaris@aegean.gr (A. Mpitziopoulos), dgavalas@aegean.gr (D. Gavalas), konstant@cti.gr (C. Konstantopoulos), pantziou@teiath.gr (G. Pantziou).

1574-1192/$ – see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.pmcj.2008.06.001

trading [24], distributed information retrieval [10], network awareness [2], network & systems management [8,22], etc. Among others, MAs have found a natural fit in the field of Wireless Sensor Networks (WSNs); hence, a significant amount of research has been dedicated to proposing ways for the efficient use of MAs in the context of WSNs. In particular, MAs have been proposed for enabling dynamically reconfigurable WSNs through easy development of adaptive and application-specific software for sensor nodes [28], for separating sensor nodes in clusters [14], in multi-resolution data integration and fusion [4,19] and location tracking of moving objects [3,27]. These applications involve the use of multi-hop MAs visiting large numbers of sensors. The order in which those sensors are visited (i.e. MA itinerary) represents a critical issue, seriously affecting the overall performance. Randomly selected routes may even result in performance worse than that of the conventional client/server model; yet, this issue is not adequately addressed in these works.

WSNs applications often include monitoring and recording of sensitive information [1] (e.g. battlefield awareness, secure area monitoring and target detection). Hence, their critical importance raises major security concerns. Jamming is defined as the act of intentionally directing electromagnetic energy towards a communication system to disrupt or prevent signal transmission. In the context of WSNs, jamming is the type of attack which interferes with the radio frequencies used by network nodes [23]. In the event that an attacker uses a rather powerful jamming source, WSN communications will be disrupted. In effect, contemporary WSNs cannot take effective measures against jamming,[1] which raises a major security issue. A notable weakness of the above-mentioned MA-based data dissemination approaches in WSNs is that none takes into account the case that communication of a significant number of sensor nodes is disrupted due to a jamming attack.

The design objective of our Jamming Avoidance Itinerary Design (JAID) algorithm is twofold: (a) to calculate near-optimal routes for MAs that incrementally fuse the data as they visit the nodes; (b) in the face of jamming attacks against the WSN, to modify the itineraries of the MAs to avoid the jammed area(s) while not harming the efficient data dissemination from working sensors.

The first objective is met through the design of a novel algorithm that separates the sensor network into multiple groups of nodes, calculates near-optimal routes (itineraries) through the nodes of each group and assigns these itineraries to individual agent objects. To meet the second objective, the Processing Element (PE) uses the JAM algorithm [29] to map the jammed area(s) and identify the problematic nodes.[2] Next, it executes queries in specific time intervals so as to be informed as soon as they resume function. Assuming that not the entire WSN is affected, the MAs are scheduled so as to bypass the jammed nodes. Instead, they visit nodes close to the jammed area(s) that are not affected in order to avoid the security risk and thus the collapse of the WSN. If the number of jammed nodes is below a specific threshold, JAID only modifies the pre-jamming scheduled itineraries ('connects' the cut-off nodes to jam-free nodes) to increase the algorithm's promptness and minimize the itinerary scheduling cost. Otherwise, JAID re-constructs the agent itineraries excluding the jammed area(s).

Admittedly, jamming avoidance exhibits several common aspects with fault tolerance in the sense that jammed nodes may be perceived by the PE as – temporary – node failures. However, jamming involves specific symptoms and characteristics that differ from those of typical node failures. Jamming may be looked at as a situation that involves massive number of node faults or communication disruptions within a specific network area. On the other hand, sensor node failures due to energy depletion or communication disruptions due to RF unit failures follow a random temporal–spatial distribution pattern.

Another important feature of JAID is the suppression of data taken from sensors when the corresponding readings have spatial/temporal similarities. For instance, an MA may calculate and store the average temperature recorded from each sensor over a monitoring period (temporal data suppression) or the minimum humidity recorded by sensors deployed in an area of interest (spatial data suppression). Data suppression gives JAID the capability to conserve valuable energy resources since the MAs carry smaller amounts of data. Namely, the more load of data the MAs carry the more energy is required for agent migrations (transmission) [30], which seriously affects the overall network lifetime. JAID uses spatial–temporal suppression [25] to reduce communication cost and maximize the network lifetime.[3]

The remainder of the paper is organized as follows: Section 2 reviews works related to our research. Section 3 discusses the design and functionality of our heuristic algorithm for designing near-optimal itineraries for mobile agents performing data fusion and security tasks in WSNs. Simulation results are presented in Section 4, while Section 5 concludes the paper and presents future directions of our work.

[1] A considerable percentage of the nodes deployed in contemporary WSNs are ZigBee [34] and IEEE 802.15.4 [9] compatible and use Direct Sequence Spread Spectrum (DSSS) modulation. However, these protocols have not been originally designed taking radio jamming into account. WSN nodes design also presents the same problem. Other types of widely utilized motes such as Mica-2 [5] are even more susceptible to jamming since they use a single frequency (433 MHz) for communication. The problem of avoiding or defending jamming attacks is very complicated and demands the use of complex and high-cost techniques (e.g. Frequency Hopping Spread Spectrum (FHSS), hybrid FHSS-DSSS, specialized antennas) [16]. The above reasons suggest these techniques as inappropriate for WSNs wherein the node manufacturing cost is a major issue.

[2] Simulation results in [29] in a WSN composed of 121 sensor nodes proved that the mapping activity varies from 1.5 s for moderately-connected networks to just over 5 s for the largest jammed region. This is fast enough to allow a reasonable prompt response to jamming.

[3] Silberstein et al. [25] provide the definitions of spatial and temporal suppression and their possible combination: (a) spatial suppression: a node suppresses its reading if it is identical to those of its neighboring nodes; (b) temporal suppression: if a node's reading is not changed since the last transmission, it does not have to report to the sink (the sink can use its previous reading as the current reading); (c) spatial–temporal suppression: combination of spatial and temporal suppression (if readings do not change, they should not be reported, otherwise, if the relationship between neighboring nodes remains the same, some reports may still be suppressed).