Computers and Electrical Engineering 000 (2017) 1–15

Contents lists available at ScienceDirect

Computers and Electrical Engineering

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

Towards scalable and load-balanced mobile agents-based data aggregation for wireless sensor networks☆

Govind P. Gupta[a,*], Manoj Misra[b], Kumkum Garg[b]

[a] Department of Information Technology, National Institute of Technology, Raipur 492010, India
[b] Department of Computer Science & Engineering, Indian Institute of Technology, Roorkee, India

ARTICLE INFO

Article history:
Received 30 November 2016
Revised 20 October 2017
Accepted 23 October 2017
Available online xxx

Keywords:
WSNs
Data aggregation
Mobile agents
Itinerary design
Bloated state problem

ABSTRACT

In recent years, mobile agents have been employed for collaborative processing in wireless sensor networks (WSNs), where agents migrate from node to node, progressively aggregate data and return to the sink with the results. However, when an agent migrates through its itinerary, its size linearly grows and it may become bloated in size. We show through mathematical analysis that the existence of the bloated state problem in mobile agents-based data aggregation causes increase in node's energy consumption as well as response time. To solve the bloated state problem, we propose a scalable and load-balanced scheme for mobile agents-based data aggregation. The proposed protocol is evaluated through simulation experiments under different network scenario and compared with some well-known existing schemes. The proposed scheme shows improved performance in terms of energy consumption, response time and network lifetime.

1. Introduction

1.1. Background

Recent advances in embedded micro-electro-mechanism system (MEMS), digital electronics and wireless communication technologies have enabled the development of small size, low-cost, multifunctional sensor nodes with sensing, computation and wireless communication capabilities [1–2]. These sensor nodes can sense various physical attributes such as temperature, sound, vibration, pressure, motion or pollutants, etc [2].

Most existing wireless sensor networks use the traditional client/server computing model for data dissemination towards the sink node and do not support self-adaptive application, because they execute statically-installed application code, which limits their flexibility [3]. In addition, this model exhibits unsatisfying performance when a network is composed of many sensor nodes, i.e. it suffers from the scalability problem [3–5]. In the client/server model, sensor nodes work as clients and transfer data to a server working as a sink node. If the transferred data-size is large, there will be a lot of network traffic in the WSN. As a result, a fair amount of collisions occur in the network and retransmission of these collided packets adds to the energy consumption of the sensor nodes [6]. It also increases the duty cycle and communication overhead for the aggregator nodes. In order to solve the flexibility and scalability problems of client/server model, some researchers have proposed the mobile agent computing paradigm for data dissemination in WSN [3,6]. In this paradigm, sensor data stays at local sites, while the data aggregation code is transferred to the data source sites. Mobile agents (*MAs*)-based *WSNs* support

☆ Reviews processed and recommended for publication to the Editor-in-Chief by Associate Editor Dr. M. H. Rehmani.

* Corresponding author.
E-mail addresses: gpgupta.it@nitrr.ac.in (G.P. Gupta), mmisra.it@nitrr.ac.in (M. Misra).

https://doi.org/10.1016/j.compeleceng.2017.10.020
0045-7906/

Please cite this article as: G.P. Gupta et al., Towards scalable and load-balanced mobile agents-based data aggregation for wireless sensor networks, Computers and Electrical Engineering (2017), https://doi.org/10.1016/j.compeleceng.2017.10.020

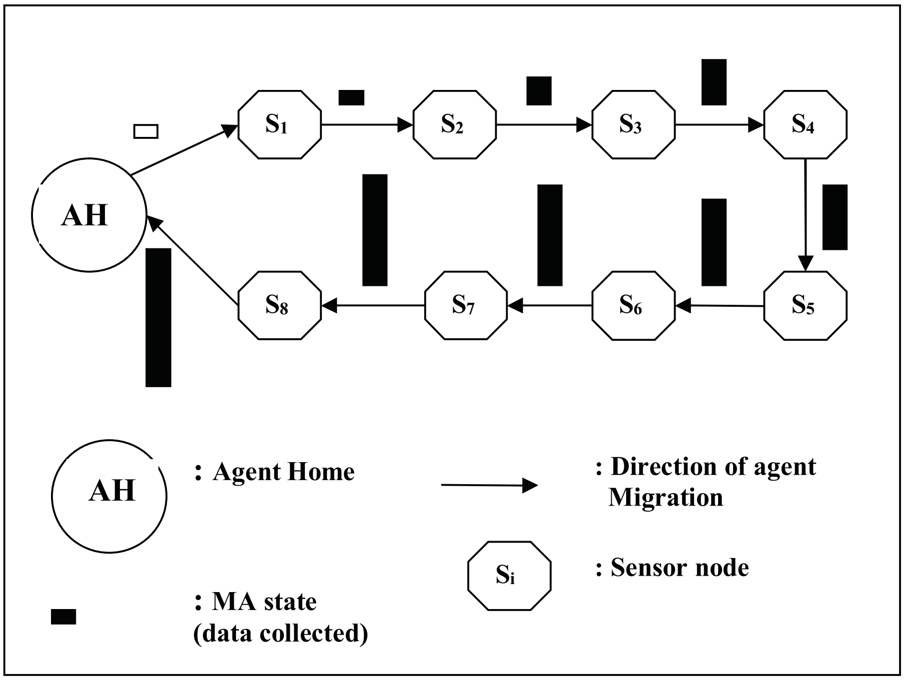

Fig. 1. Illustration of bloated state problem of the *MA* [8].

self-adaptive applications because they are deployed with no pre-installed applications [3]. Instead, users dispatch *MAs* that are autonomous programs with the ability to move from node to node, perform the assigned task and fetch the results back to the sink node. An *MA* may be defined as an entity that contains a data space to carry collected data and an itinerary which can either be fixed or dynamically determined, based on the current network status and the application code [3,6,7].

Most of the existing *MAs*-based data aggregation protocols use a static itinerary which is determined centrally at the sink [3–7]. The most notable drawback of these protocols is that an agent may not move along its itinerary due to node or communication link failures [3–6]. Also, each agent needs to carry a pre-computed itinerary list which increases its size. Moreover, these protocols suffer from the bloated state problem [8] because they do not impose any restriction on the agent payload size. When an agent collects aggregated data and migrates to the next node, its size grows because of accumulation of aggregated data at each visited node. Due to this, the agent may become bloated. A bloated *MA* suffers from migration delay and consumes more energy [8].

Fig. 1 illustrates the bloated state problem of the *MA* after sequential migration of the *MA* after data aggregation and collection from each sensor node S_i. It can be observed from Fig. 1 that an agent home (*AH*) node dispatches the *MA* for data aggregation task. An *MA* visits a sequence of eight sensor nodes (i.e. S_1- S_8). In the execution state, an *MA* contains two main containers, one for code and another for collected data. The code carried by the *MA*, allows it to perform data aggregation task with the local data. However, collected data by the *MA* from each S_i, represents payload of the *MA*. As the *MA* moves from S_1 to S_8 according to agent migration method, it collects the aggregated data from each node S_i. Without loss of generality, if an *MA* collected m bytes of data from each node S_i, size of the *MA* after visiting k nodes would be $(C+m\times k)$ bytes [8]. Here C is size of code carried by an agent. In this manner, size of an agent is bloated to $(C+m\times k)$ bytes, when it reaches back to the *AH*. In other words, we can say that size of an agent is susceptible to the data collected from each S_i as well as length of the itinerary. Once the agent is bloated, its migration to the next node consumes more energy and becomes reasonably slower. Consequently, the network lifetime may become significantly lower and response time may become high.

1.2. Main contribution and organization of the paper

In order to solve the bloating state problem of the *MA* as discussed in Section 1.1, we present a Scalable and Load-balanced Mobile Agents-based Data Aggregation (SLMADA) protocol, where the itinerary of the agent is dynamically decided at each hop. In order to prevent the bloated state problem, SLMADA sets a maximum data payload limit for the agent and utilizes the agent cloning approach, similar to the method proposed in [9]. The agent makes a clone of itself according to its payload size. SLMADA helps the *MA* to change its itinerary dynamically according to the current state of the network.

The main contributions of this paper are summarized as follows:

1. A bloating state problem in the mobile agents-based *WSNs* is investigated.