Mobile Crowdsensing with Mobile Agents

(JAAMAS Extended Abstract)

Teemu Leppänen[1], Jóse Álvarez Lacasia[2], Yoshito Tobe[3], Kaoru Sezaki[2], Jukka Riekki[1]
[1]Department of Computer Science and Engineering, University of Oulu, Finland
[2]Institute of Industrial Science, University of Tokyo, Japan
[3]RealWorld Communication Laboratory, Aoyama Gakuin University, Japan
teemu.leppanen@ee.oulu.fi[1], sezaki@iis.u-tokyo.ac.jp[2], yoshito-tobe@rcl-aoyama.jp[3]

ABSTRACT

Mobile crowdsensing applications can be designed as multi-agent systems with campaign-specific roles, realized as mobile agents, and role-based interactions. Mobile agents facilitate decentralized and autonomous crowdsesing campaign execution that takes into account dynamic resource availability in the system. Mobile agents address issues in collaborative campaign execution, campaign monitoring, data collection, data analysis and handling participant-related issues. A software framework is presented that seamlessly exposes campaigns and mobile agents as Web resources. Real-world evaluation shows comparable results with existing crowdsensing approaches but with smaller energy consumption. In-network data sharing introduces insignificant overhead.

Keywords

Multi-agent systems; Mobile agent; Mobile computing; Web services

1. INTRODUCTION

Urban phenomena can be addressed by involving humans in participatory sensing networks that rely on their mobility patterns, intelligence and social interactions. A software framework runs the campaign in participants' devices and coordinates the participants to collaborate towards campaign goals. Data are then analyzed centrally in backend systems. The variety and scale of mobile crowdsensing applications introduce challenges for system design, development and deployment. Finding suitable participants, that either opt-in to campaigns or are selected opportunistically if they meet the campaign requirements, is a challenge. Campaign execution and collected data quality is another challenge as participants' actions and interactions can not be controlled. Privacy protection has to be emphasized as the collected data can reveal the personal behaviors of the participants.

In the work [1], crowdsensing campaigns are designed as multi-agent systems (MAS) with campaign-specific roles and role-based interactions that are realized as mobile agents. This provides decentralized and autonomous campaign execution and coordination, where dynamic resource availability in the opportunistic network of participants' devices is considered by the mobile agents. While executing their roles, mobile agents consider participant's recruitment, data quality issues, privacy and compensations for the participants. Real-time data sharing enables to distribute campaign execution load. Mobile agents in a MAS facilitate concurrent execution of multiple campaigns.

Appears in: *Proceedings of the 15th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2016), J. Thangarajah, K. Tuyls, C. Jonker, S. Marsella (eds.), May 9–13, 2016, Singapore.*

2. MOBILE AGENTS IN CROWDSENSING

Campaign-specific roles embrace their associated tasks, inputs and outputs, which facilitates cooperation between roles. The role-based tasks of mobile agents include data clean-up, feature extraction, event detection, data quality assessment, privacy protection and exposing data and the task results for real-time sharing. Campaigns can be executed by injecting mobile agents into the system, where they operate autonomously, negotiate their task execution, react to changes in their environment and share their results. This enables crowdsensing online model, where sensing tasks arrive during runtime and multiple campaigns can be executed simultaneously in the same set of devices.

The sensing parameters in smartphones can be controlled by mobile agents. Manual, automatic and context-aware sensing modes are supported. Mobile agents control the sensing task based on the specified context, such as location or detected event. Different representations of the data with task execution metadata can be uploaded to the campaigner for monitoring purposes.

Participants are recruited based on their availability, location, activity, events and history. Participants' behaviors in the campaign are monitored by the mobile agents and suggestions can be made to change the in-situ data collection to eliminate unwanted behaviors. Participants' reputations are calculated by the agent for promoting suitable participants for the campaigners. Fair compensation for their resource usage is calculated for the participants by the agents.

Mobile agents assist in maintaining participant privacy by enforcing personal privacy constraints, e.g. excluding sensitive locations. With in-network data processing, mobile agents expose only selected data features to the campaign.

3. SOFTWARE FRAMEWORK

The software framework [1] is based on resource-oriented architecture, where the resources are participating devices, integrated sensors and physical components, sensor data and

agents' task results. HTTP is utilized as the universal communication protocol, which integrates campaigns and mobile agents seamlessly into the Web. Each resource is accessed through an URL containing its name and address. HTTP methods are used to access resources and to control sensors and physical components in the participating devices. A set of agent interaction protocols based on HTTP are introduced in [1] to enable mobile agent operations and conversations in the MAS.

In this kind of highly mobile distributed system, a resource directory (RD) component encompasses the real-time resource availability. Each device joins the system by registering its resources to the RD and when resource availability changes, the device updates its description in the RD. Now, campaigners can perform runtime lookups for find available resources. It is assumed that the campaigner controls the campaign through a Web service.

Figure 1 illustrates the framework and shows how crowdsensing campaigns are conducted in the framework. The devices register themselves into the RD (1). The campaigner locates resources for the campaign through resource lookups into the RD (2), injects mobile agents to execute the campaign (3) and monitors campaign execution through the results returned by the mobile agents (6). Mobile agents autonomously migrate in the system (4) and collaborate to execute their tasks (5).

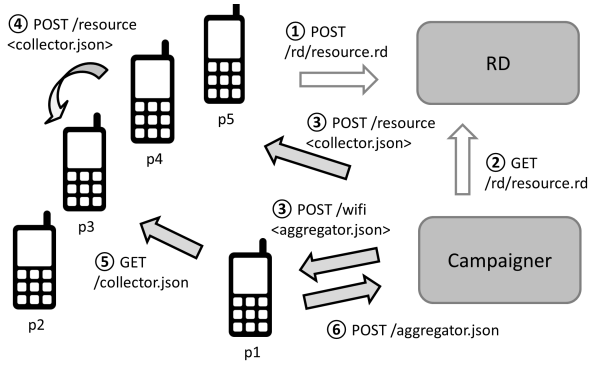
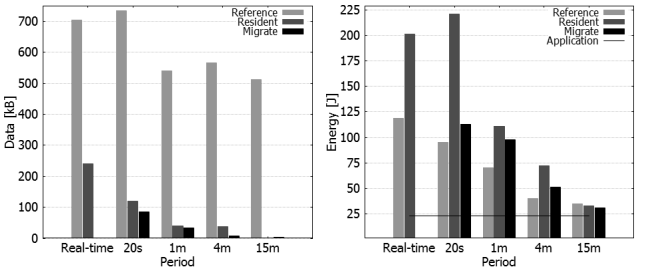


Figure 1: Crowdsensing software framework.



(a) Transmitted data (b) Total energy consumption

Figure 2: Real-world evaluation results.

4. REAL-WORLD EVALUATION

An extensive set of simulations was conducted in [1] to study different aspects of the mobile agent based crowdsensing. The total campaign execution energy was found to be lower and the amount of transmitted data to be reduced in comparison with existing crowdsensing approaches. Real-time information sharing introduces insignificant overhead to the energy consumption. Real-world evaluation was conducted in the software framework with Android smartphones in [1], hosting a mobile agent execution environment (EE) application, with a campaign that detected participant flocks from Wi-Fi scan operation data. Energy consumption data and the amount of transmitted data were then collected from the participating smartphones. Different agent migration and data upload periods where considered, as shown in Figure 2. The energy consumption of the EE application alone is also shown (Application).

The real-world campaign was implemented with two different designs. The designs were compared against a reference campaign without mobile agents, where the EE uploads raw data to the campaigner periodically. In the first design (Resident), four collector mobile agents migrate into the devices p2-p5 (Figure 1) at the join time and process collected sensor data in real-time in the device until the end of the campaign. Data from these agents is processed in the device p1 by an aggregator mobile agent before uploading it to the campaigner. This design demonstrates in-network data processing and sharing. In the second design (Migrate), a collector mobile agent migrates between the devices p2-p5 and processes their collected raw data each time the device is visited. An aggregator mobile agent operates in device p1 similarly as in the first design.

We observe that mobile agent based campaigns significantly reduce the total amount of transmitted data, which contributes towards less communication and less data processing in the campaign. Considering energy consumption, we observe that mobile agent based campaigns consume less energy across the different upload and migration periods in comparison with the reference campaign. The Migrate campaign consumes less energy than the Resident campaign, due to large chunk of data being processed and more results shared at once. Within the same period, reference campaign energy consumption is similar with the Migrate campaign energy consumption, even though it includes in-network data processing and data sharing. Also, the EE implementation suffers from high mobile agent execution overhead as explained in [1]. To address this issue, the effects of different overheads were considered in the paper. Overall, this suggests that mobile agent based campaign execution is the most beneficial with infrequent agent migrations.

5. CONCLUSION

Mobile agents based crowdsensing campaign implementation in a MAS facilitates autonomous and decentralized campaign execution. Mobile agents embrace sensing and other tasks while distributing their execution. Mobile agents enable multiple simultaneous campaigns running in the same set of participant devices. With mobile agents, campaigns can be operated more energy efficiently than the existing crowdsensing approaches and the amount of transmitted data reduced, while providing comparable results.

REFERENCES

[1] T. Leppänen, J. Álvarez Lacasia, Y. Tobe, K. Sezaki, and J. Riekki. Mobile crowdsensing with mobile agents. *Auton. Agent Multi Agent Syst.*, pages 1–35, 2015.