P2P Multi-agent Data Transfer and Aggregation in Wireless Sensor Networks

Elhadi Shakshuki, Sajid Hussain, Abdur R. Matin, Abdul W. Matin
Jodrey School of Computer Science
Acadia University
Nova Scotia, Canada
{Elhadi.Shakshuki, Sajid.Hussain, 073717m, 073720m}@acadiau.ca

Abstract

Wireless Sensor Networks (WSNs) enable pervasive, ubiquitous, and seamless communication with the physical world. This paper presents P2P multi-agent data transfer and aggregation system architecture in WSNs. The architecture includes four types of agents: interface, query, routing, and data acquisition agents. The interface agent interacts with the users to fulfill their interests. The routing agent is responsible for energy efficient data transfer. The query agent facilitates the collaboration between the interface and routing agents, and is responsible for creating optimized plans to achieve their desired goals. Both interface and query agents are placed at the resource-enriched base station because they require computation intensive operations. The data acquisition agent is responsible to acquire, filter, and format the sensor data. Sensor nodes have limited energy and computing resources. Thus, we create proxy agents for query and routing agents to act as peer agents with similar capabilities. The proxy query agents receive query execution plans from query agent, and proxy routing agents receive routing plans from routing agent. This paper provides the agents' architecture and design that enable them to coordinate and communicate with each other to transfer and aggregate data in WSNs.

Keywords*: Agents, WSN, learning, Routing.*

1. Introduction

Wireless sensor network (WSNs) is a modern paradigm for data sensing and monitoring. The goal of any type of data sensing and monitoring is to fulfill the queries generated by the users. This process of sending queries, processing, and retrieving the relevant and desired information can be efficiently accomplished using WSN; this makes WSN as an essential part of most diverse applications such as military, security, habitat monitoring, industrial automation and agriculture. The application of WSN is composed of small sensor nodes that are low-cost, low-power and multifunctional. These small sensor nodes are commonly known as motes and communicate in short distances [1]. They have the capability of monitoring the physical entities such as temperature, light, motion, metallic object, and humidity [2].

One of the major requirements of WSNs is to reduce the power consumption of limited-powered sensors. Sensors are normally deployed over a large geographical area and consume energy during communication.

Therefore, there is a need for efficient mechanisms and techniques for routing, transporting, transferring and aggregating of data from one node to another [3], which will increase the network life time [6,12]. However, due to limited computing and storage resources, the sensors are not equipped with an enriched operating system that can provide energy efficient resource management. Hence, the application developers are responsible to incorporate low-level energy efficient communication and routing strategies. This unique combination of characteristics in sensor networks, deployment environments, and largely diverse applications has recently generated a lot of interest to many researchers [5, 8].

Our proposed solution is to utilize agent-based approach as a design paradigm. Towards this end, this paper describes a multi-agent system architecture that acts as a mediator between the user and the WSN environment. This work is an ongoing research and builds on our previous work [4, 10]. The main focus of this paper is to provide the architecture, functionalities and design of each individual agent. The agents of the system have the ability to interact with each other and collectively accept queries from the user and return the desired results, communicate with the sensors in the environment, assign the heads of all clusters, learn about the status of all sensors, analyze the decisions, and act autonomously.

The remainder of this paper is organized as follows. Section 2 discusses data transfer and aggregation system architecture. Section 3 provides a brief description of the agents' functionalities, architecture, design and implementation. Section 4 presents some results. Finally, section 5 provides the main contributions of this research and outlines our goals for the future.

2. System Architecture

The proposed architecture is designed to assist users to efficiently monitor and retrieve the desired information from wireless sensor networks. Each agent is autonomous, intelligent, coordinated, and able to communicate with other agents to fulfill the users' needs.

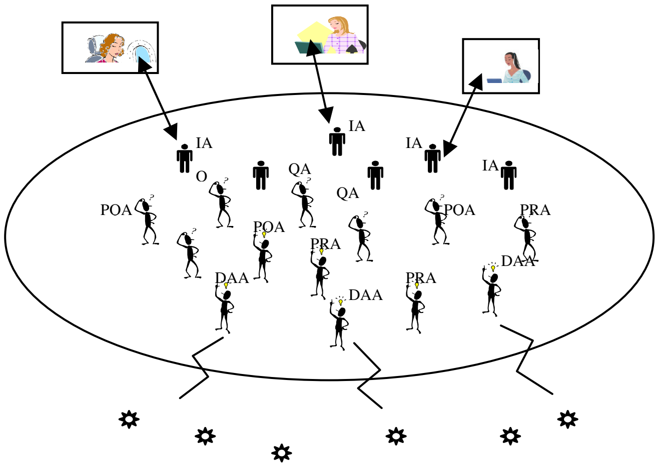


Figure 1: System architecture.

Figure 1 shows the four types of agent functionality in the system architecture. In this architecture, the Interface Agent (IA) receives user queries and requests and creates a SQL-like statement for Query Agent (QA). Since both IA and QA require computation intensive operations, they are placed at the resource-enriched base station. QA receives different user queries, creates optimized query execution plans, and sends the optimized plans to Routing Agent (RA). QA provides an application specific optimization; for instance, multiple identical queries can be served by a single query execution plan. On the other hand, RA is responsible for energy efficient data transfer, where optimization is based on network conditions, such as node density, clusters formation, and available energy. As sensor nodes have limited energy and computing resources, we create proxy agents: Proxy Query agent (PQA) and Proxy Routing agent (PRA) that act as peers for QA and RA respectively [10]. QA provides query execution plans for PQAs and RA provides data transfer plans for all PRAs. PQA receives sensor data from Data Acquisition Agent (DAA), which is responsible to acquire, filter, and format the sensor data.

3. Agents' Architecture

The architecture of all the agents is based on the agent model proposed in [9], as shown in Figure 2. Although all agents of the system have similar architecture, each agent exemplifies a particular architecture to reflect its functionality. In this model, an agent has a set of executable components and knowledge components. The executable components are represented by rectangle shapes and include: problem solver, learning, assignment, knowledge update, and communication. The knowledge components are represented by oval shapes and contain the information about the WSN environment such as number of clusters and nodes, other agents' models, goals that need to be satisfied, and the possible solutions generated to satisfy each goal. The problem solver component provides the agent with the capability to infer about its knowledge in order to generate the appropriate solution plans to satisfy a goal. The learning component provides the agent with the capability to learn; it uses the transmitted data and run machine learning techniques to provide optimized configuration of the network for efficient power consumption [11]. The assignment component allows the agent to delegate goals to other agents of the system and assigns the responsibility for achieving them. The knowledge update provides the agent with the capability to have up-to-dated information about all the elements of the environment as well as learning about the other agent's capabilities. The communication component allows the agent to exchange messages with the user and/or other agents using predefined message formats.

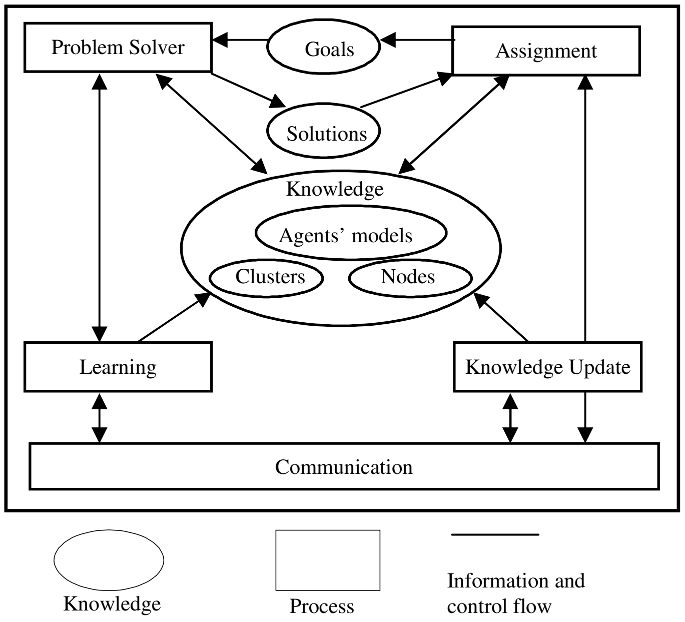
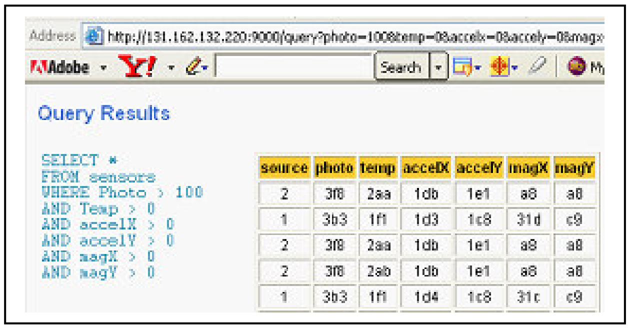


Figure 2: Agent architecture.

3.1. Interface Agent (IA)

The main responsibility of the interface agent is to allow the users to interact with the system. The interface agent is the front-end of the system and responsible for fulfilling the users' requests, by receiving queries, processing them, and delivering results. The interface agent provides graphical user interfaces (GUIs) to the user, and the services to query the sensor network. An example of a GUI interface is shown in Figure 3.


Address
http://131.162.132.220:9000/query?photo=100&temp=0&accelx=0&accely=0&magx
Adobe
Search
Query Results

```
SELECT *
FROM sensors
WHERE Photo > 100
AND Temp > 0
AND accelX > 0
AND accelY > 0
AND magX > 0
AND magY > 0
```

source	photo	temp	accelX	accelY	magX	magY
2	3f8	2aa	1db	1e1	a8	a8
1	3b3	1f1	1d3	1c8	31d	c9
2	3f8	2aa	1db	1e1	a8	a8
2	3f8	2ab	1db	1e1	a8	a8
1	3b3	1f1	1d4	1c8	31c	c9

Figure 3: An example of an interaction window with a user.

These agents are located on computational devices such as desktops or laptops, and have a light-weighted architecture, including: problem solver, assignment, knowledge update, communication, and knowledge. The agent maintains the HTTP connection with the user through its communication component. The problem solver consists of three processes, including receiving, request-handler, and service.

The *receiving* process is a daemon and responsible for receiving all user requests submitted through the communication component at a well-known TCP/IP port. The *request-handler* parses the incoming requests, formulate the appropriate message to be sent to the query agent through the service process, and display the results to the user. For each incoming request from a user, an instance of a request handler is created; this enables the interface agent to receive concurrent requests and maintain the TCP/IP connections with the user application. The service process allows the agent to interact with the routing agents, using the assignment component. The service process sends the formulated queries to the desired routing agents. When it receives the results from the routing agents, it combines them and forwards them to the appropriate request-handler.

3.2. Query Agent (QA)

For query dissemination and efficient in-network processing query agent is introduced in the proposed system, which resides at the base station. They act as a gateway between the interface and routing agents. Since the sensor nodes have limited energy supply, the sensor nodes are equipped with proxy query agents (PQA) similar to query agent and responsible for local data processing at the sensor nodes.

The architecture of the query agent consists of the following components: problem solver, assignment, knowledge update, communication, and knowledge. These components allow the agent to generate a real-time optimum query plan for routing the data. The problem solver consists of two processes: query optimizer and query planner. The *query optimizer* process receives query packets from the IA. It continuously receives queries from the interface agent and thus forms a queue to handle all the queries. The function of the query optimizer is to handle redundant queries by either processing one query at a time or forming a single joint query. The query optimizer first checks its knowledge for the availability of the query results. If the information requested is available locally in its knowledge, the results are retrieved and sent back to the interface agent through its communication component. If the requested information is not available locally, a query is forwarded to the query planner based on the priority or location in the queue.

The *query planner* is responsible for generating a query plan for each incoming query. The query planner generates an optimum query plan by evaluating different query plans utilizing its knowledge and estimates the cost of each plan, and then updates its knowledge accordingly. The query agent aggregates the incoming data from the network nodes. The data aggregation takes place at the cluster heads and finally at the base station. It updates its knowledgebase using the knowledge update component.

3.3. Routing Agent (RA)

The routing agent main responsibility is to perform real-time energy efficient routing decisions. To achieve its objective, it uses machine learning techniques at the base station. Due to the constraint energy resources of the sensor nodes, sensor nodes are equipped with thin client proxy routing agent (PRA). Although the proxy routing agent has similar architecture to that of the routing agent at the base station, its functionality is adapted according to the requirements.

The architecture of the routing agent consists of problem solver, assignment, learning, knowledge update, communication, and knowledge. The routing agent's *learning* component uses genetic algorithm to provide an optimum network configuration for an optimum number of future transmissions [10]. First, it creates an initial population with random cluster size. The available nodes and their start energies are obtained from the knowledge. Then, several generations are created to produce an optimum chromosome. The optimum chromosome (solution) provides the complete network details: the number of cluster heads, the members of each cluster head, and the number of transmissions for this configuration. The optimum solution is stored in the agent's knowledge for future use. In addition to optimum solutions, the knowledge contains all the information about the network conditions, user applications, and user interests.

The problem solver component is a decision maker that analyzes the contents of the knowledge to choose an optimum routing plan for an optimum number of future transmissions. The problem solver component sends the decision to the assignment component. Using this component, the agent assigns the responsibilities to the sensor nodes. The communication component creates packets that are transmitted to the desired sensor nodes. The communication components of the proxy routing agents, at the sensor nodes, receive the packets broadcasted by the routing agent at the base station. The agents at the proxies record the appropriate network and cluster information in their knowledge.

The proxy routing agents transmit packets according to the given plan for a given number of transmissions. Once the current plan is completed, the problem solver component chooses new plan from the knowledge and the network is reconfigured accordingly.

3.4. Data Acquisition Agent (DAA)

The data acquisition agents are embedded in the sensor nodes and responsible for gathering the raw data and filtering the noise from the data. The architecture of this agent consists of problem solver, knowledge update, communication, and knowledge. The communication component receives requests submitted by proxy query agents and sends the filtered data back to them. The filtering is performed by the problem solver component. As soon as the query agent receives those results, it transmits them to its cluster head through its proxy routing agent. Since the data acquisition agents are implemented on the back-end of the system, which makes the network hardware independent. For example, mica2 sensors can be replaced with tmote sensors [7], without making changes to the design of the other agents of the system.

4. Application Example

To demonstrate the operation of the system in real world application, we implemented a prototype for a sensor network application using Java and mica2 motes. The mica2 motes are attached with MIB310 sensor boards. The application monitors temperature, light, and mobility in the computer science department, at Acadia University. In this implementation, the sensors are placed in rooms 403, 410, and 419 as shown in Figure 4.

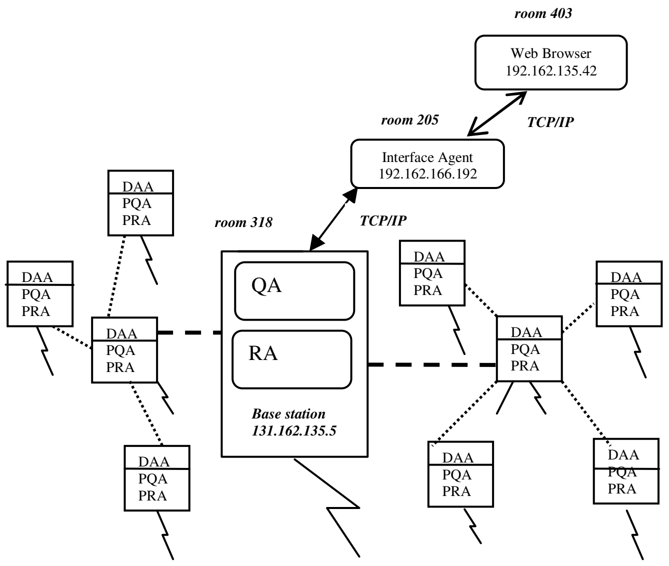


Figure 4: An application example.

One of the mica2, mote, which acts a base station, is connected to the PC located in room 318; the current IP of the base station is 192.162.135.5. Three different conditions are considered and can be described as follows: (1) four motes are placed in room 403 that has normal room condition, (2) four motes are place in room 419 where temperature is warmer, and finally (3) four motes are placed in a meeting room 410, which is cooler than the others. The IA is running on a PC with IP address 192.162.166.192 located in room 205. The receiving process of IA is managed by a web server. For each incoming user request, an instance of the request-handler process is created in a separate thread. The service process is implemented in Java as RMI server; which is available at rmi://131.162.135.56:9090.

Using the interface shown in Figure 3, a user is allowed to submit a SQL-like query. The HTML input text fields are used to enter the threshold values for the sensor readings. For example, the user in room 403 with IP 192.162.135.42 sends the query in the form of "SELECT TEMP FROM sensors WHERE TEMP > 0 AND PHOTO >100". The service process of IA forwards this query to the base station that is located in room 318.

The QA residing in base station generates a query plan and sends it to the RA; RA through its learning component generates the network configuration. For the current scenario RA generated 3 clusters, one in each of the following rooms: 403, 410 and 419. The RA broadcasts the packets through its radio for the nodes present in all the three rooms. The proxy routing agents on all nodes listen to the high signal broadcast messages transmitted by the base station. When the PRA on the cluster head receives the packets, it consequently sends them to the PRA on their member nodes, which in turn send the query packet to PQA.

Finally, the PQA gets the temperature readings from DAA that read the data and sends it back to the PQA. PQA forwards the data to the PRA which transmits the query results to the cluster heads. The PQA on the cluster head performs local aggregation of the query results and sends it to its PRA which routes the query results to the RA on the base station in room 318. The RA then sends the packets to its QA. The QA aggregates the incoming packets of the sensor nodes where the temperature is greater than 0 and photo was greater than 100; the aggregated results are send to IA and then displayed to the user.

A sample results are presented in Figure 3. The results showed the previous 5 readings received from all the sensors. It should be noted that the readings are in hex-decimal notation. In addition to that, the query can be adjusted. For example, retrieve the most recent, average, maximum, or minimum value, received from any or all the sensors.

5. Conclusions and Future Work

This paper presented a multi-agent system architecture for data transfer and aggregation in wireless sensor networks, with special focus on the agents' architectures. The system facilitates the main characteristic required to reduce the power consumption, in which the sensors are deployed in a large area with limited power. The agents work together as peers and are aware of the dynamic changes in the power of the sensors during communication and maintain efficient mechanisms and techniques for communication and routing strategies. A prototype of the system is implemented using Java and mica2 motes.

Currently, we are investigating different techniques for the election of cluster agents and compare them. In the future, we are planning to incorporate the energy level of the cluster agents in the query dissemination plan as well as incorporating other machine learning techniques and compare our results for better performance.

Acknowledgements

This project is supported in part by a grant from the Natural Sciences and Engineering Research Council of Canada, and an internal grant from Acadia University.

References

[1] F. Akyildiz and W. Su and Y. Sankarasubramaniam and E. Cayirci, "Wireless Sensor Networks: A Survey", Computer Networks, no.38-4, pp. 393-422, March 2002.

[2] D. Culler, D. Estrin, and M. Srivastava, "Overview of sensor networks", IEEE Computer, vol. 37, no. 8, August 2004.

[3] D. Estrin, R. Govindan, J. Heidemann and S. Kumar, "Next century challenges: scalable coordination in sensor networks, "in Proceedings of International Conference on Mobile Computing and Networks (MobiCom'99), Seattle, Washington, August 1999.

[4] S. Hussain, E. Shakshuki, W. Matin, "Agent-based System Architecture for Wireless Sensor Networks", in the Proceedings of The Second International Workshop on Heterogeneous Wireless Sensor Networks (HWISE 2006), in conjunction with the 20th International Conference on Advanced Information Networking and Applications, IEEE Computer Society, accepted for presentation, April 18-20, Vienna, Austria, 2006.

[5] D. Massaguer "Multi Mobile Agent Deployment on Wireless Sensor Networks", http://www.cs.wustl.edu/mobilab/

[6] V. Mhatre, C. Rosenberg, D. Kofman, R. Mazumdar, and N. Shroff, "A minimum cost heterogeneous sensor network with a lifetime constraint", IEEE Transactions on Mobile Computing, vol. 4, no. 1, pp. 4–15, 2005.

[7] J. Polastre, R. Szewczyk, and D. E. Culler, "Telos: enabling ultra-low power wireless research", in the Proceedings of Information Processing in Sensor Networks: Special track on Platform Tools and Design Methods for Network Embedded Sensors (IPSN/SPOTS), April 25-27, 2005.

[8] J. S. Sandhu, A. M. Agogino, A. K. Agogino "Wireless Sensor Networks for Commercial Lighting Control: Decision Making with Multi-agent Systems", American Association for Artificial Intelligence, 2004.

[9] E. Shakshuki, H. Ghenniwa, M. Kamel, "Agent-based System Architecture for Dynamic and Open Environments", International Journal of Information Technology and Decision Making, 2(1), pp. 105-133, 2003.

[10] E. Shakshuki, S. Hussain, A. R. Matin, A. W. Matin ''Agent-based peer-to-peer layered architecture for data transfer in Wireless Sensor Networks", The IEEE International Conference on Granular Computing (GrC), IEEE Computational Intelligence Society, accepted for presentation, May 10-12, Atlanta, USA, 2006.

[11] E. Shakshuki, S. Hussain, A. R. Matin, A. W. Matin, ''Routing Agent for Wireless Sensor Networks'', Jodrey School of Computer Science, Acadia University, TR-2005-08, 2005.

[12] N. Trigoni, Y. Yao, A. Demers, J. Gehrke, and R. Rajaramany. Wavescheduling: Energy-efficient data dissemination for sensor networks. In Proceedings of the International Workshop on Data Management for Sensor Networks (DMSN), in conjunction with the International Conference on Very Large Data Bases (VLDB), August 2004.