



A Survey of Agent Technologies for Wireless Sensor Networks

Orhan Dagdeviren, Ilker Korkmaz, Fatih Tekbacak & Kayhan Erciyes

To cite this article: Orhan Dagdeviren, Ilker Korkmaz, Fatih Tekbacak & Kayhan Erciyes (2011) A Survey of Agent Technologies for Wireless Sensor Networks, IETE Technical Review, 28:2, 168-184, DOI: [10.4103/0256-4602.72509](https://doi.org/10.4103/0256-4602.72509)

To link to this article: <https://doi.org/10.4103/0256-4602.72509>



Published online: 01 Sep 2014.



Submit your article to this journal [↗](#)



Article views: 68



View related articles [↗](#)



Citing articles: 2 View citing articles [↗](#)

A Survey of Agent Technologies for Wireless Sensor Networks

Orhan Dagdeviren, Ilker Korkmaz¹, Fatih Tekbacak and Kayhan Erciyes²

Department of Computer Engineering, Izmir Institute of Technology, 35430, Urla,

¹Izmir University of Economics, 35330, Balçova,

²Izmir University, 35350, Uçkuyular, Izmir, Turkey

Abstract

Wireless sensor networks (WSNs) do not have a fixed infrastructure and consist of sensor nodes that perform sensing and communicating tasks. The WSNs have large application spectrum such as habitat monitoring, military surveillance, and target tracking, where sensor nodes may operate distributed in highly dynamic environments. Battery-constrained sensor nodes may aggregate the sensed data, localize themselves, and route the packets in an energy-efficient and decentralized manner to enable running the applications. Agents are capable of independent and autonomous action, so that they can successfully carry out tasks that have been delegated to them, thus agent-based approaches are very suitable to apply as the solution of the problems occurring in WSNs. So far many agent-based approaches were proposed for WSNs. This paper surveys the agent technologies for sensor networks by providing a classification, objectives and costs of these approaches with the open research problems. To the best of our knowledge, this is the first study that covers the intersection of the agent technology and sensor networks from a wide perspective.

Keywords

Agent middlewares, Communication protocols, Data sampling algorithms, Energy harvesting algorithms, Localization algorithms, Mobile software agents, Mobile hardware agents, Multi-agent systems, Task assignment algorithms, Wireless sensor networks.

1. Introduction

Wireless sensor networks (WSNs), containing self-organizing and cooperating low-power sensing nodes [1] have attracted the academia and the industry due to their wide range of large-scale applications in the last decade. Battlefield surveillance, chemical contaminant detection, microclimate control, and environmental monitoring are some important application types of WSNs. As an example, many sensor nodes can be deployed in an optimized manner [2] or randomly in a forest to track the movements of a flock of animals or to detect and localize the place of a fire where the environment is dynamic. The nodes operate in a distributed and energy-efficient way to construct a network architecture [3] for forwarding the sensed data, to make localization for tracking the targeted objects [4], to maintain a fault-tolerant topology in case of malfunctions, to aggregate the sensed data for reducing the number of transmitted messages, etc.

An agent is a computer system that is situated in some environment, and which has a capability of acting autonomously in this environment to meet its design objectives [5]. Agents autonomously sense the

environment and respond accordingly. There is a need for a selection of criteria in an environment to establish an agent-based system. According to Gerhard [5], firstly, the environment has to be open, highly dynamic, uncertain, or complex. Thus agents are very suitable to be deployed in the sensor network environment, where many approaches were proposed to solve the well-known problems in sensor networks by using agents [6-8].

In this survey, we provide a detailed classification of agent technologies with the open research issues for sensor networks. To the best of our knowledge, this is the first study that surveys the agent-based solutions for sensor networks by a detailed classification. There are some other survey papers related to agent approaches for sensor networks [6-8]; however, these lack a comprehensive classification of the agent technologies.

The rest of this paper is organized as follows: in Section 2, we give the classification method of agent technologies for sensor networks. We present the approaches of agent-based technologies in Section 3. In Section 4, we explain the open research issues and finally the paper is concluded in Section 5.

2. Classification of Agent Technologies for Sensor Networks

An agent, according to its native concept of use in computer science, is abstractly a combination of computer software and data. Based on their first attitude of being autonomous, which mainly means for agents to be able to act without a human owner even in dynamic environments, and their specialized capability to move or conceptually to migrate between computers, native mobile agents are kind of software agents in their terminological sense. However, concerning the use of agent technologies with today’s ubiquitous computing paradigms, WSN applications have been using the ‘agent’ term in more various contexts. Basically, mobile agents in WSNs are referred to either mobile processes including software codes with/without states, or mobile devices with/without special hardware supports.

In this section, we classify the agent technologies for sensor networks in three categories: mobile software agents, mobile hardware agents, and sensor nodes as agents. Related surveys focus on sensor nodes as agents [6,8] and mobile software agents [7]; they dissect these subjects well but do not mention about hardware agents and do not provide a general point of view.

2.1 Mobile Software Agents

Mobile software agents used in WSNs are semantically the classical software agents, which are specifically written to run autonomously in adaptive sensor environments. Mobile software agents, as the software codes, can be used as generic agent middleware platforms embedded in sensors or they can be used as separate frameworks integrated within the sensor network application. A

mobile software agent in a sensor network is the executable process with its state and code, and it can migrate from a sensor to another one in the network. In a WSN, different mobile software agents may be constructed for individual utility tasks, which work cooperatively to accomplish the main objective of the network application.

2.2 Mobile Hardware Agents

Mobile hardware agents are specialized devices, which can traverse the network to collect information from ordinary sensor nodes for various purposes [9]. These devices are powerful hardware units of processing, memory, communication, and mobility capabilities. Mobile hardware agents can be unmanned vehicles, robots, or any other powerful devices.

2.3 Sensor Nodes as Agents

Each sensor node may be modeled as an agent, which may reason about their environment using sensor data in an autonomous manner. Therefore, they are suitable for sensor-based applications in distributed and decentralized environments and the individual agents in the network aim at achieving systemwide goals.

3. Agent Technologies for Sensor Networks

Figure 1 depicts the agent-based approaches in this section as shown. We classify the agent technologies into three main categories and detailed taxonomy of these are shown in Figure 1. Mobile software agents can be used in two groups as agent-based middlewares and applications without any agent-based middleware support. Mobile hardware approaches are grouped in three different types as communication protocols, localization algorithms, and energy harvesting algorithms. According to their

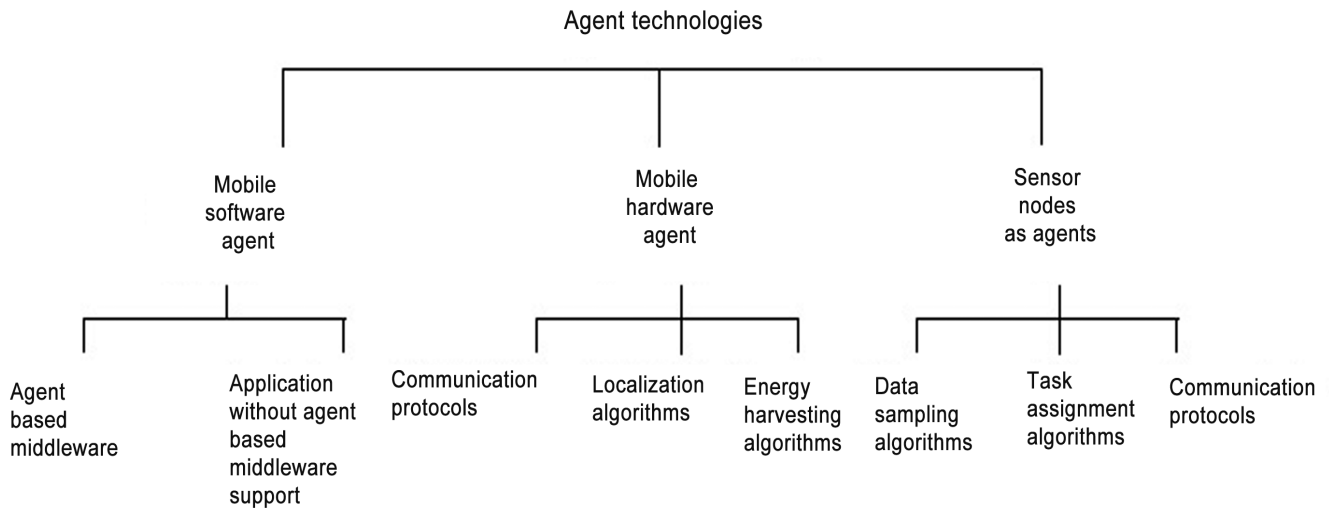


Figure 1: Classification of agent technologies for sensor networks.

individual missions, ordinary sensor nodes are modeled as agents to handle data sampling, task assignment, and communication in distributed sensor networks.

3.1 Mobile Software Agents

Mobile software agent can be seen as a special computer program that has the ability to decide on the time and the place to migrate between the sensor nodes. When the agent chooses to migrate to another node in the WSN, it logically transports its state and code to the destination and pursues its objective as keeping on executing its code on that node.

Generic working principles of mobile software agents within sensor networks are depicted in Figure 2. According to the architectural differences in WSNs' topologies, the agents' operational processes on data gathering and data dissemination in WSN may change. For example, within a cluster-based hierarchical WSN, clusterheads may instantiate the agents. On the other side, within a flat WSN, only the sink node may be able

to create an agent as shown in Figure 2. There may also be a tree-based WSN infrastructure in which the parents and/or some special aggregator nodes are the only nodes to construct the mobile software agents. In any kind of WSN infrastructure, a mobile software agent is constructed in a place with its code and initial status, and then it migrates on target areas within the WSN.

In Figure 2, a small part of a cluster-based WSN and a small part of unclustered flat WSN topologies with client/server-based and mobile agent-based communication schemas are drawn. In clustered WSNs, a mobile agent is usually created by the cluster leader to be sent to the cluster members to collect the possible information. Then, the agent may use its specific capability on specific aggregation functions by migrating between the neighbor member nodes. Only one agent may traverse the cluster territory and bring the aggregated sensed data to the leader. In flat WSN topologies, a mobile agent is usually created by the sink node and traverses the network adaptively. Finally, the agent collects and transfers the sensors' data to the sink.

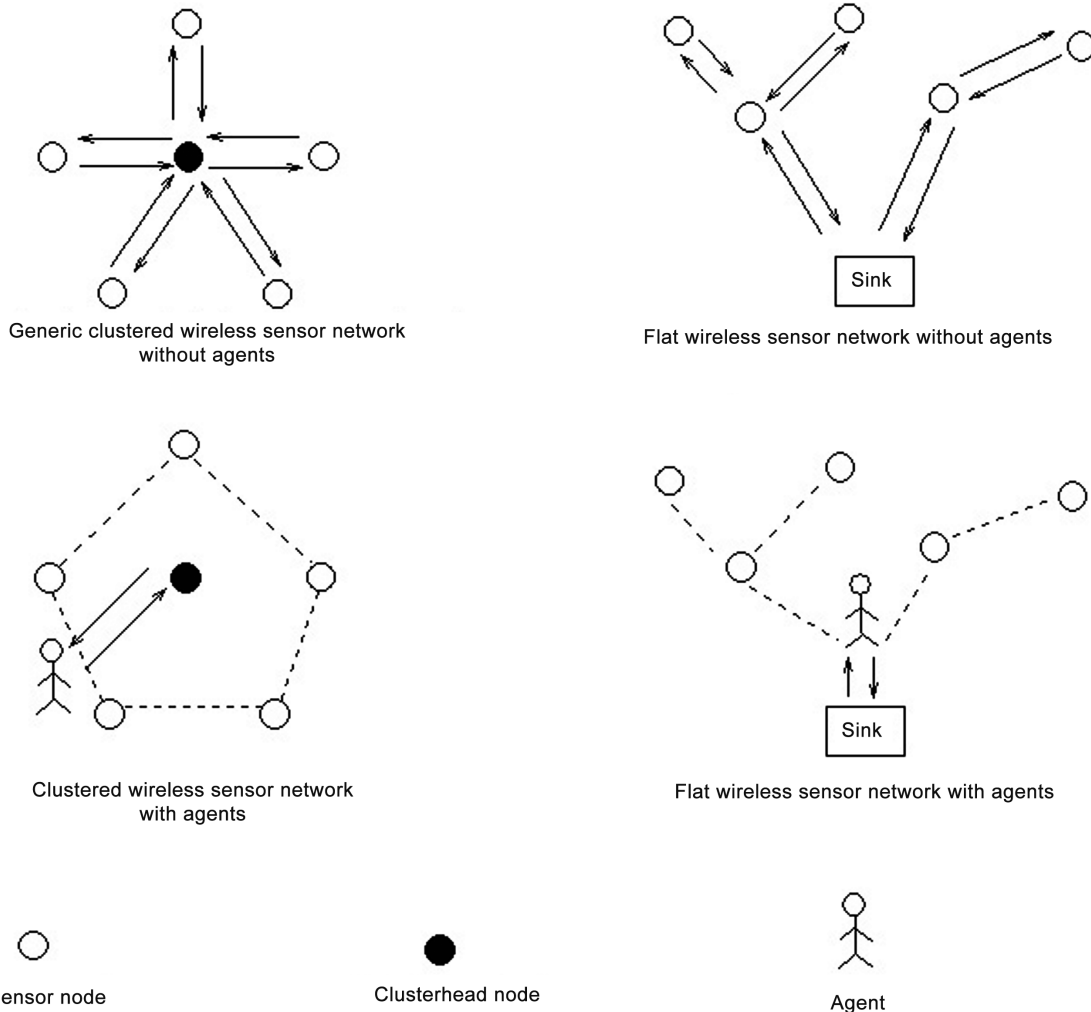


Figure 2: Mobile software agents.

If the client/server-based messaging approach drawn as in Figure 2 is used, the communication bandwidth will be consumed inefficiently compared to the agent-based approach. The mobile software agent use in WSN reduces the message traffic, and gives an opportunity to save energy. Besides, agent-based paradigms may allow more efficient memory use in sensors. Due to local memory and processor constraints in sensor nodes, to embed and to run different codes in nodes are expensive or sometimes infeasible; therefore, the agent utilization allows dynamic applications to be deployed on the same WSN [7].

The mobile software agents are used with sensor networks in two different approaches: application-integrated codes or frameworks without a middleware layer support and mobile agent middleware frameworks. The summary of these approaches is given in Table 1.

3.1.1 Mobile Agents without Middleware Support

Mobile software agents usually need middleware architecture to cooperatively work for the common WSN application. However, some approaches that use agents in the sensor networks directly use the agents without the support of generic middleware architecture. Instead, they only use a kind of a code framework or a middleware-like platform. Due to the lack of a well-defined middleware layer in this design, the software agents might be seen as functional services, the use of agents might be seen as a tool, and the implementation of such a design might just be seen as software code utilities for the WSN.

Shakshuki *et al.* [10] proposes software-agent-based directed diffusion approach to increase the node lifetime by reducing the energy costs caused by message

communication in directed diffusion approaches used in WSN. The main difference between software-agent-based directed diffusion and the usual directed diffusion without any agent use is that the first uses agent-based paradigm for the diffusion messages, whereas the latter uses a client/server-based paradigm. Shakshuki *et al.* [10] use stationary and mobile agents. A stationary agent makes its associated sensor node know its neighbors' information; on the other hand, a mobile agent visits the stationary agents and gets the related node information to decide for the optimal routing path of that node based on the neighbors' battery cost. According to their simulation results, agent-based diffusion approach is better than classic-directed diffusion in terms of connectivity, energy consumption, latency, and load balancing criteria. As those good performance results show, agent-based system suggestion of Shakshuki *et al.* [10] may be used for directed diffusion applications in WSN. However, it is not a generic middleware platform and does not provide a middleware layer to support the adaptive software agents to handle sensor applications. The authors indicate their future work so as to make their agents be realized on motes with TinyOS, which is a good roadmap to make the design be used by various WSN applications. In our view, instead of using a different kind of bundle of separate code modules, implementing the suggested design within a well-known middleware framework may also be a complementary action for agent-supported mote application designers.

3.1.2 Mobile Agent Middlewares

The mobile software agent middleware framework embedded within a sensor network is the fundamental agent-based programming platform of the network. This framework mainly provides the software migration between the sensors.

Table 1: Summary of mobile software-agent-based frameworks for WSNs

Approach	Design objectives	Pros and cons
Shakshuki <i>et al.</i> [10]	To increase the network lifetime for directed diffusion	An agent-based approach with good energy performance; but not a generic middleware
Agilla [11]	To provide a mobile-agent-based platform on TinyOS motes	A realistic middleware with strong migration in use and different applications can run simultaneously
SensorWare [12]	To create mobile control scripts and provide them with movement in WSN	Multiuser support for dynamical programming; but no agents, weak migration of scripts
Smart messages [13]	To provide a distributed computing platform for networks of embedded systems	Lightweight; but not a WSN-specific middleware
WISEMAN [14]	To combine wave system architecture with WSNs for agent use	Task coordination is focused simply by agents; however, task functionalities are left to nodes
Freitas <i>et al.</i> [15]	To build a reflective middleware for heterogeneous WSNs	Multi-agent-based reasoning support in WSNs; however, the system is targeted for WSNs including different types of devices
AgentScape [16]	To present a multi-layered semantics-ready sensor architecture	Generic sensor interface and database interface support, promising popular use after ZigBee and IEEE 802.15.4 interfaces
Others	To support code distribution in between sensors, such as Mate [17], XNP [18], Deluge [19], and Impala [20]	Utility code bundles as middleware-like frameworks without any agent-based paradigm

Agilla [11], is the most popular agent-based middleware framework for WSNs. Agilla middleware, as shown in Figure 3, can be run as a separate layer above the TinyOS operating system. With the help of Agilla middleware, one or more mobile agents that coordinate with each other are used in the network to achieve a specific goal. Agilla helps in simplifying the deployment of adaptive WSN applications in order to make implementations easily programmable and make network flexible. Mobile software agents in the system are programmed by assembly-like programming language that provides some high-level instructions to handle complex tasks. Agilla, through one instruction, can realize the agents' movement between sensor nodes by carrying only the code, which is called as weak migration, or by cloning the code and the state together, which is called as strong migration. In Agilla's main design, the new instructions cannot be integrated to a deployed network; however, Fok *et al.* declare it as a future plan [11]. New agents can be adapted to sensor nodes, or new application deployments can behave in the dynamic network environment. As shown within the layered architecture in Figure 3, Agilla maintains local tuple space and neighbor list for sensor nodes to coordinate agent communication. Multiple Agilla agents assigned for different tasks can be run simultaneously in the same WSN. Fire tracking or battlefield area application goals can be satisfied by using Agilla's efficient and reliable services.

SensorWare [12] is a middleware framework designed for wireless ad-hoc sensor networks. SensorWare framework is not defined as an agent-based implementation since no agent is used in. However, SensorWare can be seen as an agent-like system due to its use of mobile executable control scripts, which are similar to agent codes. Through those mobile scripts, the WSN can dynamically be programmed by multiple users by injecting predefined instructions. Even the system is

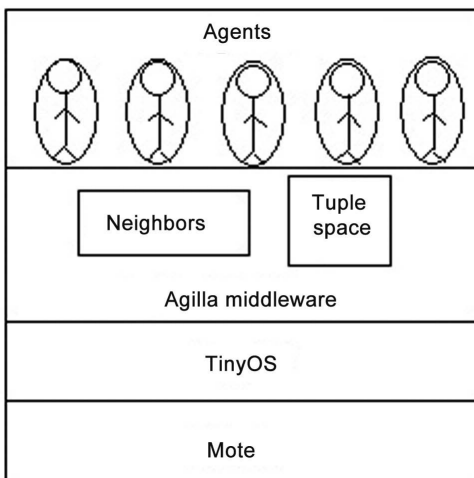


Figure 3: Agilla.

not a mobile agent middleware; its design objective is similar to agent middleware due to SensorWare's capability on migration of the scripts to different nodes, to enable dynamic deployment of distributed tasks in a WSN. Nevertheless, due to the lack of a mobile agent middleware paradigm, that similarity is limited with only weak migration, therefore, only the script code modules with no state concept of agent-based paradigm are transferred in WSN.

Smart Messages [13] is a distributed computing platform for networks of embedded systems. As it is declared by Kang *et al.* [13], it uses the ideas of migrating execution and mobile agents. A Smart Message (SM) in the system is a distributed program, which migrates with its code and execution state. A virtual machine supports the cooperative execution of SMs even in dynamic network configurations. In SM programming, the migration is explicit to the programmer, and so the programmer knows the time and the target to migrate the SM. The SM migrations are lightweight and use a shared memory. However, in the implementation of SM transfer phase, Transmission Control Protocol is used for a reliable communication between neighbor nodes. On the receipt of the proper acknowledgment, SM is transferred from the source to the destination. In our opinion, this mechanism may not be suitable for WSNs for the following reasons: after the transfer is completed, SMs create a new thread above the virtual machine within the destination node, which may not be effective for resource-constrained sensor networks. The SMs is neither a WSN-specific middleware nor a mobile agent middleware.

WISEMAN, later as named by Chen *et al.* [7], is described by Gonzales *et al.* [14] with its initial architectural design concepts. WISEMAN, which supports not only the SensorWare's capability of coarse-grained task coordination but also the Agilla's functionality on resource-limited nodes, seems to be a powerful agent-processing platform [7]. Gonzales *et al.* [14] combines wave system architecture with WSNs and uses WAVE programming language for agent migration implementation in WSNs. The authors propose a coarse-grain approach to keep the agent code compact and to make the agent just focus on the distributed task controls. The application-specific task definitions are left to be embedded in sensor nodes. A general view of a mobile software agent use in WISEMAN middleware as depicted in Figure 4 illustrates that some tasks among the three application-specific missions are loaded in the nodes and the distributed coordination of related tasks is controlled by the agent. In our opinion, this architecture is simple and has good performance for mobile agent-based distributed cooperation in WSNs, however, it may not be suitable for general WSN applications, because some complex functionalities for general-aimed

applications may need large amounts of resources to be stored, and this may be infeasible for all nodes. On the other hand, WISEMAN middleware may logically be a true choice if an application-specific design is chosen for the whole sensor network architecture, and if the nodes are locally capable to store and run the application-specific predefined tasks possibly available on the mote operating system.

Heterogeneous sensor networks as well as homogenous ones for general applications may need to adapt to different application scenarios by the node change on configuration and mission during runtime, because, adaptability is a basic point for the long lifetime and proper deployment of network in highly dynamic environments. Freitas *et al.* [15] collect user requirements as missions and translate them to network environment or node capabilities for reasoning about this information. Multi-agent approach is applied during the adaptation and reasoning process of the system. Freitas *et al.* [15] also uses high-level mission description language (MDL) to specify required desires and constraints. This language has an advantage over Agilla's approach by representing mission goals of agents in the system that reason around the environment. For example, an area surveillance system should find unauthorized vehicles in the restricted area. When a ground sensor sends a signal about an unauthorized vehicle, Unmanned Aerial Vehicles (UAVs) that carry visible-light cameras should detect the related vehicle. However, the change of weather conditions may prevent UAVs to operate properly. At that moment, middleware starts to run for adaptation of mission completion and infrared cameras continue to complete the mission instead of visible-light cameras. In our opinion, this middleware may not be used fully in homogenous sensor network infrastructures; however, its contribution on combining the reasoning concept through multi-agent approach with WSN applications is important.

Harman *et al.* [16] proposes a multi-layered semantics-ready sensor architecture, AgentScape, which extends

a multi-agent platform by supplying generic services to access sensors and store sensor data. By this approach, different sensor and database types could be considered by user agents. AgentScape architecture supports the virtual sensor-based data with the combination of real sensor signals. The main components within AgentScape middleware are shown in Figure 5 as follows: AgentScape operating system (AOS) kernel, agent server, host manager, location manager, and web service gateway. Agents in this architecture reside in defined locations and communicate with other agents and services that are external software systems hosted by the middleware. The design issue of this architecture includes a middleware kernel that implements primary mechanisms, high-level middleware services that implement agent-specific operations, and external directory services. Agent-servers, host and location managers, and a web service gateway are defined as agent-specific operations. Hosts run on locations that have location managers. Location manager manages location's hosts and every host executes a host manager that manages middleware components on itself. Each host can run one or more agent servers to host agents. Agent servers provide a runtime environment for agents. The goals of AgentScape used within sensor environment are to supply a generic sensor interface for agents' access, to provide a generic database interface for storing, and to combine different sensors' data for determining them in other ways as agent-web service coordination. In our opinion, if future plans of the ZigBee and IEEE 802.15.4 interfaces' integration to the current system in [16] are realized, the system may be popular in use for many WSN applications due to its ready-to-use condition.

There are also some other applications or middleware-like frameworks, which are not based on agent technologies but are used as the initials of distributed

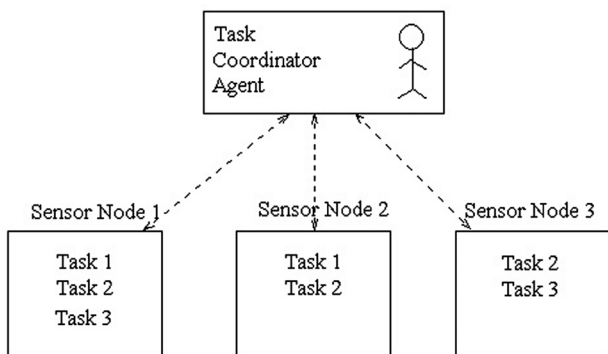


Figure 4: WISEMAN.

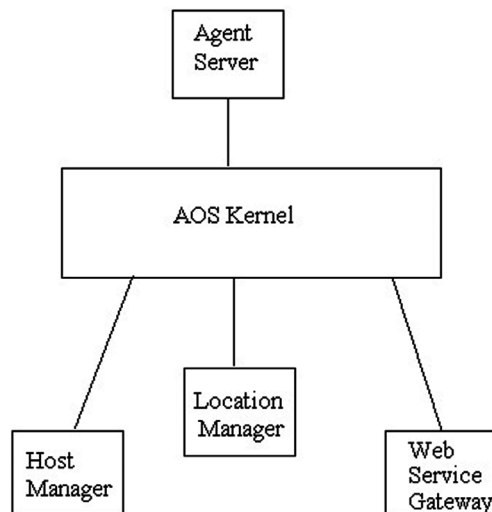


Figure 5: AgentScape.

code execution platforms in WSNs. We do not include them in any of our three main agent technology categories classified here. For example, Mate [17] is a communication-centric virtual machine designed for sensor networks as an interpreter running on nodes.

Although Mate is not a middleware, Mate programming language is the inspiration point of Agilla's assembly-like instructions. Another example is XNP [18] platform, which is not agent-based but is used for distributing codes on TinyOS-supported sensor motes. Deluge [19] is another middleware that presents the code transferring without the use of agent paradigm. As a last example, Impala [20] also provides the native code distribution over WSN without agents. The main drawback of these code bundles is the lack of ability to run different applications simultaneously in the sensor network since the codes should be installed into all nodes of the WSN [11]. In our opinion, that is an important reason to switch to agent-based paradigms for code migrations in WSNs.

Table 1 summarizes the mobile software agent technologies described in Section 3.1, where the main design objective of each approach is pointed out briefly and the pros and cons are briefly explained based on the agent paradigm used in the approach together with its possible use in sensor networks.

3.2 Mobile Hardware Agents

The proposed approaches based on mobile hardware agents in the literature can be divided into three classes as: communication protocols, localization, and energy harvesting algorithms. The summary of these approaches

is given in Table 2. We will explain these approaches in the following sections.

3.2.1 Communication Protocols

Tong *et al.* proposed an architecture for large-scale low-power sensor networks referred to as sensor networks with mobile agents (SENMA) [9]. A predominant concern for sensor networks is energy efficiency. Frequent multi-hop relaying of the sensed data may consume the battery of the nodes quickly. Besides, the communication processing overhead caused by routing and Medium Access Control (MAC) protocols of a flat ad hoc network may be very large. Moreover, in flat ad hoc networks, authors stated that most of the transmissions are between low-lying antenna where signal decays at the fourth power of distance. SENMA targets to solve these problems by introducing mobile hardware agents as the receiving terminals in data collection. SENMA provides MAC and physical layer designs for communication between sensor nodes and mobile agents. In this design, the mobile agents execute sophisticated algorithms, whereas the MAC and physical layer mechanisms executed by ordinary nodes are simple. A slotted time division multiplexing is used in SENMA for communication between mobile agents and sensor nodes. The mobile hardware agents transmit a beacon at the beginning of each slot to achieve synchronization with sensor nodes. Ordinary nodes listen to this beacon and transmit their packet based on opportunistic ALOHA. The mobile agents can be flying airplanes where there may be a free space between sensor nodes and mobile hardware agent devices. In this case, the signal only decays as the second power of distance during transmis-

Table 2: Summary of mobile hardware agent technologies

Approach	Design objectives	Pros and cons
SENMA [9]	Constructing an architecture including medium access and physical layer	Simple medium access and physical layer for sensor nodes but cost of agents with advanced hardware for algorithm execution and mobility support
C-SENMA [21]	Clustered approach of SENMA	Simple medium access and physical layer for sensor nodes but cost of agents with advanced hardware for algorithm execution and mobility support
MULEs [22]	Constructing an architecture for data collection by mobile hardware agents with random paths	Simple data collection mechanism but cost of hardware support for MULEs and high data routing latency to the base stations
Diehl <i>et al.</i> [23]	Improves Shah's approach [22] by defining effective data collection paths	Energy-efficient data collection mechanism by hardware agents but cost of agents with advanced hardware support to calculate and move to effective paths for data collection
McLaughlan and Akkaya. [24]	Agents move and locate to cover the sensor nodes by maintaining a k -hop independent set	A dynamic communication infrastructure but cost of agents with advanced hardware for mobility support
Melodia <i>et al.</i> [25]	An event-driven clustering framework where the problem is formulated by integer programming	An event-driven dynamic communication infrastructure but cost of agents with advanced hardware for algorithm execution
Pathirana <i>et al.</i> [26]	Localization with an agent by applying extended Kalman filter on received signals	Simple and accurate localization but cost of an agent with an advanced hardware for executing complicated algorithms
Lee <i>et al.</i> [27]	Localization with an agent by processing the images of sensor nodes	Simple and accurate localization but cost of a hardware agent with image processing capability
Rahimi <i>et al.</i> [28]	Energy harvesting by energy producing agents	Network lifetime is guaranteed but 40% of the nodes should be energy-producing agents

sion as stated in SENMA. In Figure 6a, the communication between ordinary sensor nodes and the mobile hardware agents as proposed in SENMA is depicted. Each message transmission is drawn in Figure 6a as dashed lines. The ordinary nodes may transmit sensed data to the specialized flying devices by executing a lightweight communication protocol. The authors showed that this approach yields an asymptotic improvement from $O(n^2)$ to $O(n)$ in the energy consumption of packet transfers in dense sensor networks.

Even though collection of the sensed data by the mobile hardware agents in SENMA significantly decreases the energy consumption of the communication in a sensor network, further reductions may be possible. Lotfinezhad *et al.* improved the SENMA as C-SENMA by introducing clustering operation [21]. In C-SENMA, the ordinary nodes are grouped into clusters such that they only communicate with the nearest clusterhead. The data aggregation and communication to the mobile hardware agents are carried by clusterheads. Cluster formation is triggered by the mobile agents periodically. Each node selects itself as a clusterhead with a fixed probability of p_c . After a node is chosen as clusterhead, it broadcasts an advertisement message to its neighbors. Each ordinary node joins into the cluster of the nearest clusterhead by approximating the distances based on received signal strength of the advertisement messages. After the completion of the clusters, Time Division Multiple Access is used for the intracluster communication. When a clusterhead collects all data from its members, it performs a data aggregation and contends for the reachback channel to send the data to the mobile agent. The authors showed that by using this scheme, the network energy expenditure can be reduced more than 80% compared to the unclustered approach. An example operation of the C-SENMA is illustrated in Figure 6b. In SENMA shown in Figure 6a, all the nodes contend for the

reachback channel, on the other hand in C-SENMA, only the clusterheads participate in the contention as shown in Figure 6b. Thus the number of message transmissions between sensor nodes and mobile agents, shown with dashed lines in Figure 6b, is reduced to the number of clusters. For example, when the number of clusters is \sqrt{n} for a network with n nodes, the transmission number in clustered approach is asymptotically improved from $O(n)$ to $O(\sqrt{n})$ compared to the unclustered approach.

Shah *et al.* proposed an architecture to maintain the connectivity in sparse sensor networks [22]. The architecture has three layers in which randomly distributed sensor nodes collect data from the environment in the bottom layer. The middle layer is composed of the mobile agents, which are defined Mobile Ubiquitous LAN Extensions (MULEs) wherein they move randomly and communicate with the sensor nodes to receive the data collected from the environment. MULEs forward these data to the top layer which includes access points. The authors showed that their architecture can maintain the connectivity on sparse networks. Diehl *et al.* argued that if mobile agents move on a planned path instead of randomly, then the energy consumption of the sensor nodes will decrease [23]. The authors stated that given the restricted travel time due to the limited energy budget, the mobile agent may not able to reach the location of every sensor node to minimize the power consumption of the sensor nodes and to maximize their lifetimes. To achieve this goal, two new schemes, clustering-based and minimum spanning tree-based path planning, are proposed. Clustering-based scheme groups the nodes into clusters and then the agent visits the center of each cluster. In minimum spanning tree-based scheme, the agent starts with a minimum spanning tree with location of all sensor nodes and then dynamically removes edges according to its residual energy. Figure 7 shows the effect of the path planning to the data collection in sensor networks. The

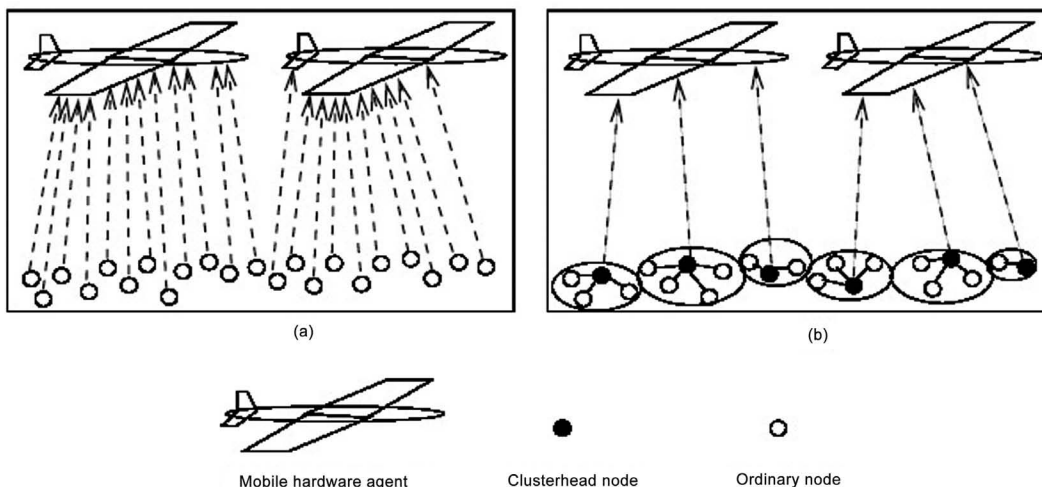


Figure 6: (a) SENMA (b) C-SENMA.

mobile agent moves in a random walk pattern in Figure 7a and cannot cover eight sensor nodes. With a clustering-based path, the number of uncovered nodes by the agent is reduced to two as shown in Figure 7b. Intuitively, it can be seen in Figure 7 that the mobile hardware agent following a random path may move more than the one following a predetermined path in order to cover all of the sensor nodes.

A mobile hardware agent can move from one location to another for providing a communication infrastructure. McLaughlan *et al.* proposed a *k*-hop independent dominating-set-based algorithm to place agents in such way that the coverage of agents is maximized and the data gathering is minimized [24]. The coverage is defined as the total number of sensors under the action ranges of all agents divided by the total number of sensors in the region. A dominating set (DS) (*S*) is a subset of *V* such that each vertex not in *V* is adjacent to at least one vertex in *S*. A *k*-independent dominating set (*k*-IDS) is a set of dominators such that any two dominators in the set are not adjacent of each other and a domantee node is at most *k*-hops away from at least one of the dominators [24]. Authors proposed a distributed algorithm to construct the *k*-IDS in which, some of the nodes advertise themselves as dominators by broadcasting *DOMINATOR* message to their *k*-hop neighbors based on a computed probability. This probability depends on the number of received *ALIVE* messages from the neighbors. If a node receives a *DOMINATOR* message, it will be dominated by the originator of this message. The dominators are also clusterheads in this scheme. The nodes at the border of the clusters broadcast *BORDER* messages. The message complexity of the distributed

algorithm is $O(s)$, where *s* is the number of dominators. A 2-IDS is depicted in Figure 8 where mobile agents are the dominators. In Figure 8, there are three dominators and 28 ordinary nodes. Any ordinary sensor node in this figure is at most two hops away from at least one of the three dominators. For example, Node 9 can communicate with Node 29 via the path Node 9 → Node 3 → Node 29. Through their simulations, the authors showed that their algorithm provides up to 40% increase in coverage and 25% decrease in end-to-end delay compared to random-proximity-based clustering algorithm.

Other mathematical models than graph theory can be used by agents for constructing a communication infrastructure. Melodia *et al.* proposed an event-driven clustering framework in which cluster formation is triggered by an event so that clusters are created dynamically to react to the event optimally [25]. The authors divided the problem into two sub-problems: Sensor-agent coordination problem and agent-agent coordination problem. The objective of the sensor-agent coordination problem is to find the data aggregation trees from all sensor nodes that reside in the event area to the appropriate agents and it is formulated as an integer linear program. The objective of the agent-agent coordination problem is to select the best agent to perform appropriate action in the event area. This problem is formulated as a mixed integer non-linear program and an auction-based localized solution is presented.

3.2.2 Localization Algorithms

Localization in sensor networks involves determining the location of the sensor node based on other nodes'

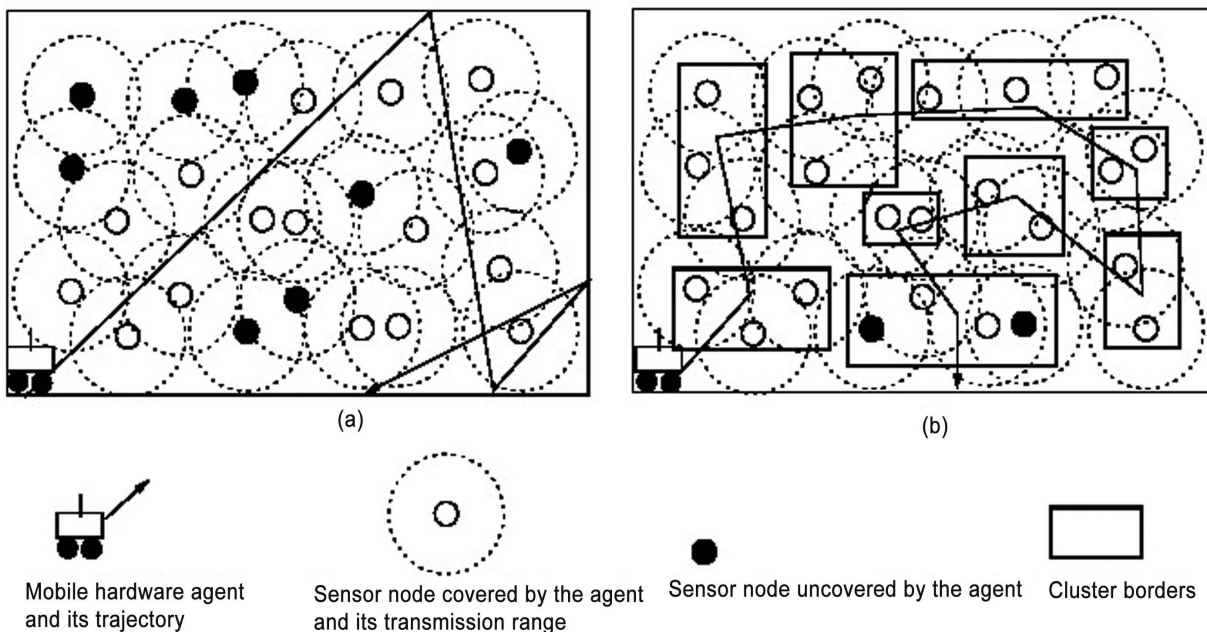


Figure 7: Data collection paths for mobile agents.

coordinates [29]. Pathirana *et al.* [26] proposed that a mobile hardware agent can traverse the network to collect the signals from sensor devices to localize them as shown in Figure 9a. Pathirana *et al.* stated that the computational requirements for localization of sensor devices can be reduced by using mobile hardware agents. Besides, they stated that since the agent receiver is mobile, the fading noise in received signal strength measurements can be statistically eliminated over a period of time. Moreover, the localization technique can be more sophisticated since processing is performed by the agent. They proposed applying a robust extended Kalman filter-based state estimator for node localization. The localization accuracy of their algorithm is approximately 1 m measured from the real experiments. Their algorithm performs better than the previous algorithms where the localization accuracies are within 3 m. Lee *et al.* proposed a vision-based technique for localization in sensor networks

[27]. Different than the Pathirana *et al.*'s approach in Figure 9a, motion plane is parallel to the sensor nodes plane and the agent device is equipped with a camera as shown in Figure 9b. The mobile hardware agent uses the camera for processing the images of sensor nodes. Lee *et al.* studied two scenarios in which the agent knows its global coordinates or does not have this information. Besides, they proposed neighbor discovery schemes and used this information in discovering the topology of the network.

3.2.3 Energy Harvesting Algorithms

In many proposed approaches for sensor networks, it is assumed that the batteries cannot be replaced or recharged [30]. Rahimi *et al.* proposed that some of the specialized nodes are autonomously mobile, allowing them to move in search of the energy, recharge and

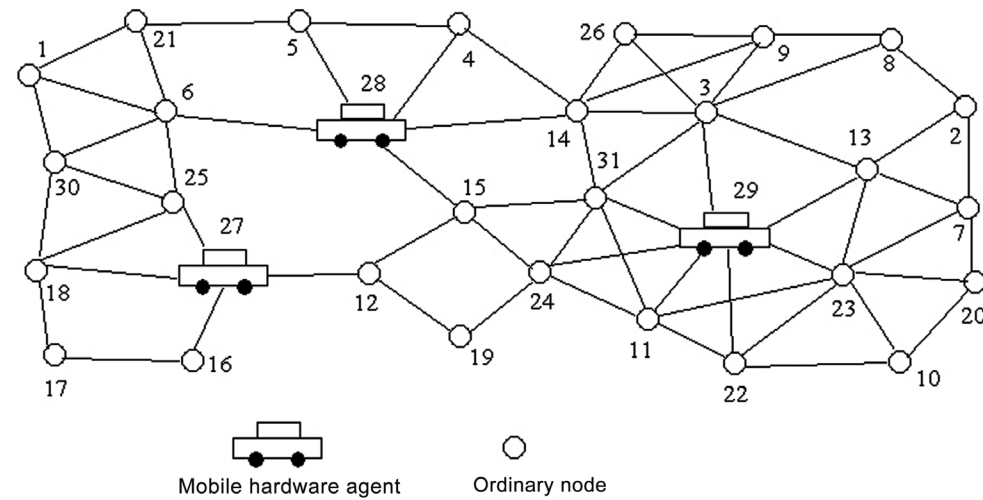


Figure 8: Mobile hardware agents for providing communication infrastructure.

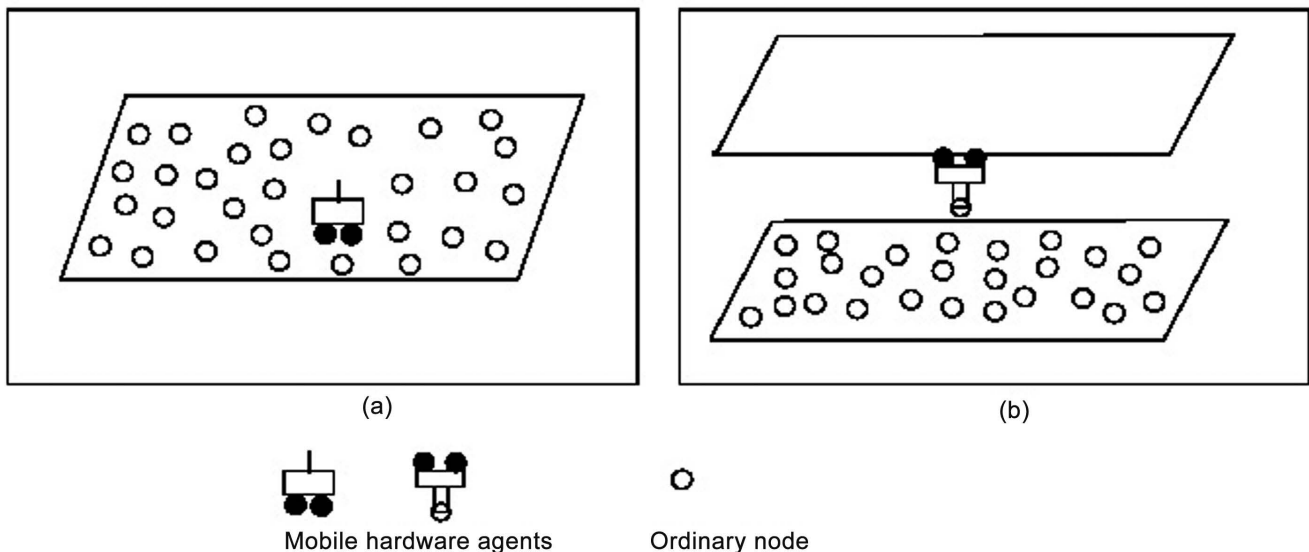


Figure 9: Mobile hardware agents for localization.

deliver the energy to immobile, energy-depleted nodes [28] wherein this term is called energy harvesting. Rahimi *et al.* assume that the energy consumption of the network is non-uniformly distributed. Therefore, energy producer mobile agents should recharge the consumer nodes in an efficient manner to prevent the energy starvation in some portion of the network. To solve this problem, the sensing area is divided into energy cells as shown in Figure 10, where energy cell is defined as the territory of each agent. The number of static sensor nodes, the rate of their energy consumption, and available environmental energy determines the number of agents per cell. The mobile energy producer agents may use solar energy for recharging themselves. It is shown that if 40% of the total nodes are producers, the network lifetime is guaranteed. An example operation is depicted in Figure 10 in which the sensing area is divided into six cells wherein eight nodes are energy-producing mobile hardware devices. These agent devices are responsible for harvesting energy to the nodes in their cells. In the PlantCare project by

Sigurdsson *et al.*, mobile hardware agents are used for energy producers [31]. The authors stated that the power to the passive nodes can be delivered in a variety of forms: A sufficiently agile robot could replace weak batteries with fresh ones or inductive recharging could be used.

Table 2 summarizes the mobile hardware agent technologies described in Section 3.2, wherein the main design objectives together with their pros and cons are given for each approach.

3.3 Sensor Nodes as Agents

This section covers the approaches which model sensor nodes as agents. Table 3 shows the summary of these approaches.

3.3.1 Data Sampling Algorithms

Sensor nodes may adjust their sensing (sampling)

Table 3: Summary of sensor nodes as agents

Approach	Design objectives	Pros and cons
Backcasting [33]	Adaptive data sampling by communicating with the base station	Energy-efficient data sampling but centralized approach controlled by the base station
SORA [34]	Adaptive data sampling by Wellman’s market-oriented programming	Energy-efficient data sampling but centralized approach where the price of each action is set by the central coordinator
USAC [35]	Adaptive data sampling and routing	Energy-efficient data sampling but confidence interval value remains static throughout the systems’ operation
Kho [32]	Decentralized adaptive sampling	Energy-efficient data sampling but transmitting rate and schedule are fixed
Kho’s routing [36]	Fixed and flexible routing	Flexible routing delivers more packets but does not scale well
Stranders <i>et al.</i> [37]	Coordination approach for mobile sensors to collect data	Mobile sensor nodes quickly find the events but when the number of nodes increase, movement paths to necessary locations decrease
Le <i>et al.</i> [38]	Task assignment	Solution to task assignment problem by using agents but sensors should be static and missions are independent

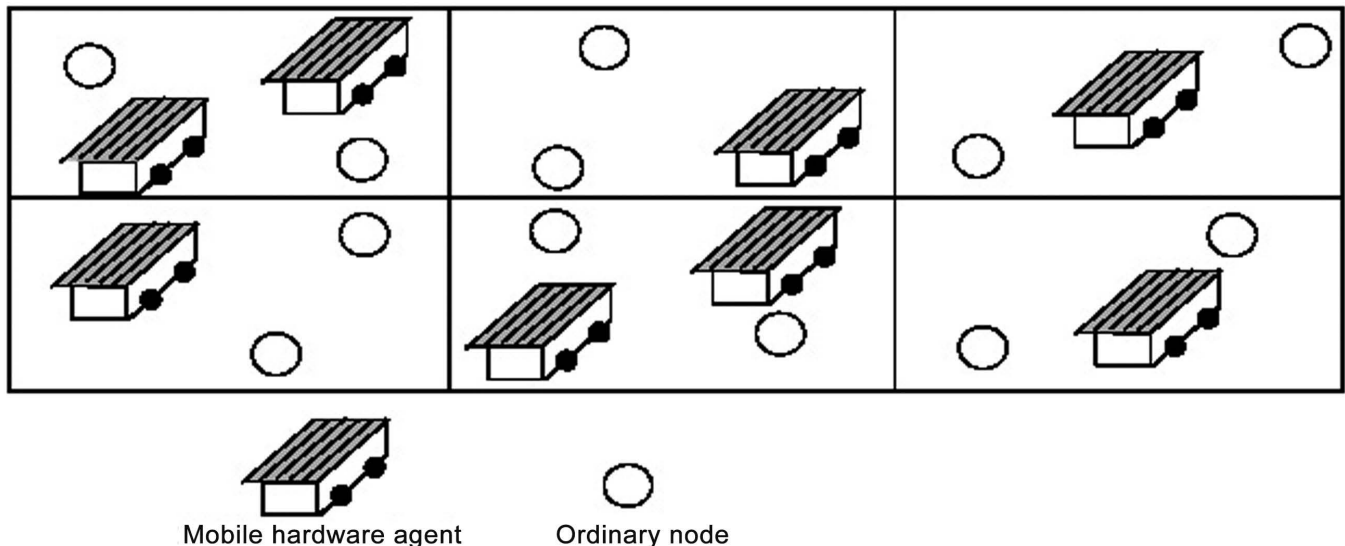


Figure 10: Mobile hardware agents for energy harvesting.

capabilities in order to improve their energy management. A node may schedule its sampling interval based on its past set of observed data and the set of data that it believes it will observe, so as to achieve the network's goals [32]. Willett *et al.*'s [33] backcasting adaptive sampling algorithm needs a small subset of sensor nodes to communicate with a base station. The base station may activate additional nodes in order to decrease the error ratio. The key idea behind the backcasting is that many sensor nodes may not need to be activated since they measure the correlated data. The authors analyzed the potential of adaptive sampling in the context of field estimation wherein it is assumed that total of n sensor nodes are uniformly deployed. It is shown that in a smooth field with an array of 100×100 sensors, this method can reduce the energy consumption by nearly a factor of 10 with the same accuracy achieved when all sensors are activated.

Mainland *et al.* [34] determined self-organizing resource allocation (SORA) as an approach that nodes make data sampling to maximize their profit by taking energy-efficient actions. This approach is inspired by Wellman's work on market-oriented programming [39]. In this approach, each useful action, like recording meaningful sensor data or forwarding messages, has a price. The price of each action is set by the central coordinator who determines the global behavior of the network. Each node has a budget that may be consumed by the actions that it can take. In SORA, each node monitors its local state and its price vector (a list including price of each action) to select the best action that maximizes its utility. The authors showed that SORA consumes less energy than static and dynamic scheduling.

Padhy *et al.* [35] defined a utility-based sensing and communication model (USAC) that included sensing and communication protocols by the decentralized control of the system. USAC has two main features: a mechanism for adaptive sampling, and a routing protocol. In the adaptive sampling approach, each node in USAC arranges the frequency of sampling rate according to observations. If the predicted data of observations are out of the confidence interval, sampling rate increases. Otherwise, sampling rate decreases to yield energy efficiency.

Kho *et al.* [32] proposed an adaptive sampling algorithm for energy-constrained networks. They modeled the adaptive data sampling problem as person-task assignment problem. The aim of person-task assignment problem is to assign a set of people to do set of tasks where each person takes a certain amount of time to do a certain task. They solved the problem by using binary integer programming and they developed a decentralized algorithm for FloodNET application.

The algorithm provides individual nodes to make local decisions based on its observations. The authors showed that this approach has approximately 44.5% less uncertainty error than a uniform non-adaptive approach.

3.3.2 Communication Protocols

Kho *et al.* [36] developed two different decentralized algorithms to solve the problem of communication in wireless visual sensor networks where each node acts as an agent. In the first algorithm (fixed routing), route from nodes to the base station is not changed. In the second algorithm, route between nodes and base station is not fixed. To sense, route and forward messages, three types of messages are used. First type of message contains node sampling data. Other messages called coordination messages are meta-data messages and control messages. Meta-data messages include content of visual data with the number of samples during the production of data. Control messages allocate resources to supply efficient energy consumption. Sending control messages before actual data messages increases necessary data rate for the base station. In fixed-routing, there is only one predefined unique path between a node and the base station. In flexible-routing, an arbitrary path has been selected for routing the data by the nodes. There are levels for each node to indicate the distance between node and the base station. Nodes belonging to the third level have three-hop distance to the base station. Each node sends its data to a node that is nearer to the base station. Therefore, a tree-structured routing mechanism has been constructed in the network. Transmission, routing costs, and process time during sampling have been compared between these approaches. Flexible routing transmits data twice according to fixed routing. However, communication and computation during sampling costs are 100 times more for flexible routing. The more nodes the network has, the more efficiency obtained by the fixed routing.

Stranders *et al.* [37] proposed a coordination mechanism that enables the sensors to learn the environmental parameters online from their collected information. To observe the environment dynamically, Gaussian processes that are used to inference about functions have been used for location awareness of sensors and replan paths for sensor nodes. Sensors have to be continuously adapted to the environment. Previous observations have to be collected to infer about the environment. Time-dependent variations are also important for momentarily changing environments such as the increase of temperature in an environment. Authors used autonomous learning agents to handle these problems. These agents measure the environment and exchange observation information with other agents in a decentralized manner. The collective information

helps agents to decide their path planning and by the share of location information, agents move to the less informative parts of the environment.

3.3.3 Task Assignment Algorithms

Sensor networks obtain multiple sensors over a wide area connected through a communication network. Coordination of usage of critical resources (sensors, communication cost, and power) is a main concern in environmental monitoring of sensor networks [40]. In this kind of environments, mission-based and organizational issues have to be managed in the existence of conflicts in the system. Every sensor node should know that it is a small part of a larger organization and that it may need to satisfy more global goals at the expense of its own local goals [41]. As stated in [15], different missions can be applied to sensor environment but these missions may conflict in dynamic sensor environments, therefore, sensor resources have to be matched with related missions in a proper manner.

The coordination of sensors by running sensors as agents helps to realize mission-based tasks [38]. Allowing sensors to be shared and reassigned between different tasks also increases power savings. To verify this reality, Le *et al.* proposed to decompose missions as tasks and these tasks can share different assets by optimizing the sensor resources according to their reassignment [38]. These sensors, named sensor-agents communicate with each other to cooperate for assigning mission-specific tasks and related assets in an optimum manner. A mission includes tasks where each task has a specific type, own location, operational range and requires a number of sensor types. A task is completed if all the sensor types demanded by it are available and the mission is completed after the completion of all tasks. The overview of this process is given as a flowchart in Figure 11.

Table 3 condenses the approaches using sensor nodes as agents as described in Section 3.3.

4. Open Research Issues

In this section we will discuss the open research issues for the agent technologies in sensor networks.

4.1 Mobile Software Agents

Mobile software agents should be used with a realistic middleware support. Therefore, the research issues on mobile software agent use in WSNs may follow two different road maps: the first is the work to build a well-performing middleware design, and the second is the work to design mobile agent-based applications for WSNs.

In middleware projects, instead of the agent-like frameworks, more realistic agent-based middleware platforms can be preferred to work on. From the point of software mobile agent use within a middleware, rather than initiating a different mobile agent-based platform implementation, one of the available ones such as Agilla can be chosen to develop new features within this platform. Agilla middleware may be investigated to provide large number of concurrent WSN applications. There may also be focused on possible correlations of data to be gathered by different agents, which run for different applications in the same network environment.

Among the other middleware platforms, WISEMAN can also be chosen for application-specific WSNs. The factors and the metrics for the application-specific sensor network design may be researched in details on the basis of agent-based platform use. The special trade-off criteria of these metrics for different kinds of applications in different kinds of WSN topologies can be evaluated through different scenarios.

AgentScope may be a promising mobile agent-based middleware platform for various WSN applications. Its IEEE 802.15.4 and ZigBee interfaces are promising and there is a potential to link its agent use with various WSN applications. The conceptual and practical feasibility or infeasibility about bringing pre-deployed WSN applications and AgentScope system together can be researched. If feasible and also flexible, the procedures and/or methodologies for transforming the usual WSN

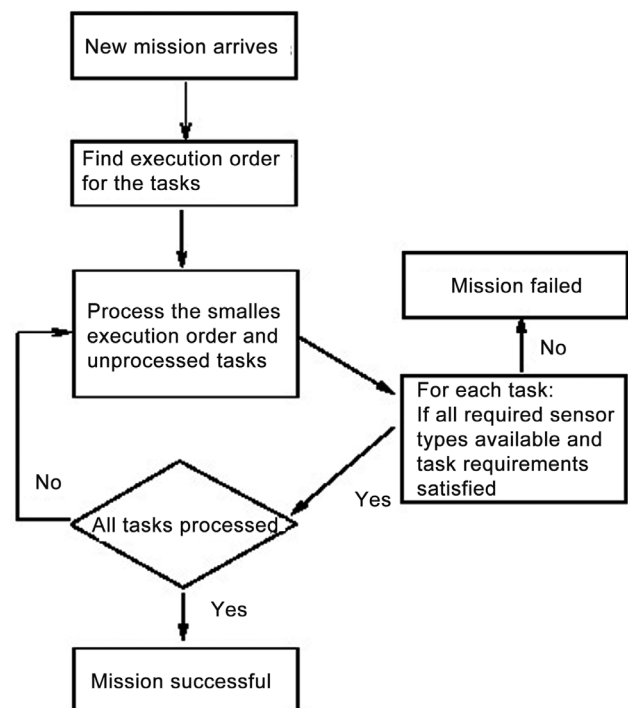


Figure 11: Mission-based task operation.

scenarios to novel agent-based WSN technologies may be the future work to investigate.

From the design phase to the deployment stage of mobile agent-based applications for WSNs, some heuristics could be searched for some specific applications to find whether an optimization is possible. If it is possible according to the application requirements and objectives, to abstract and consider the scenario as a distributed constraint optimization problem (DCOP) may be remarkable since distributed sensor networks are resource-constrained and agents have so far been used for constraint optimizations.

4.2 Mobile Hardware Agents

4.2.1 Communication Protocols

Mobile hardware agents may run sophisticated algorithms so that we may shift the computationally intensive jobs from sensor nodes to the agents. The algorithms executed on the agent side may be improved to give better services to the sensor nodes. For example in C-SENMA, it is assumed that sensor nodes consume uniform energy and clusterhead selection is based on rotation among the nodes. However, there may be hot spots in the sensor network due to randomness of the occurring events. Besides, clusterheads may fail at any moment leading to the loss of data. The C-SENMA architecture, which is an improved version of SENMA, can be further improved with the energy-efficient and fault-tolerant clusterhead selection. In Pathriana *et al.*'s and Lee *et al.*'s localization approaches, the mobile hardware agent may obtain all coordinates of the sensor nodes where this information with the knowledge of transmission range can be used to construct a weighted graph. Mobile agent may execute clustering algorithms or other topology control mechanisms centrally on this graph. Other than these additional services, which are not suitable for distributed implementation in resource constraint sensor networks, services such as security primitives based on asymmetric cryptography [42] may be provided by the mobile agent.

Mobile hardware agents can be clusterheads as proposed in McLaughlan *et al.*'s *k*-IDS algorithm. They may move to different locations and stay there to maximize the number of covered sensor nodes and to construct an efficient communication backbone. This ability of the mobile hardware agents can be useful to solve other distributed graph theoretical problems. A Steiner tree is a minimum spanning tree that contains all nodes of a predefined set of nodes, adding other nodes as required and constructing a Steiner tree is an NP-complete problem, but has a constant approximation factor [43]. Steiner trees are useful to

construct energy-efficient backbones and to provide multicast infrastructures. Design of the approximation algorithms to solve the Steiner tree problem, where the predefined set of nodes can be the sensor nodes and the other set of nodes can be the mobile hardware agents is an open research issue.

4.2.2 Localization Algorithms

The localization approaches proposed in [26,27] are executed on a single mobile hardware agent. Thus these approaches are prone to single point of failures which means that the localization service given by the agent completely stops when the agent fails. Fault-tolerant solutions may be proposed to overcome this situation. One solution to this problem is the introduction of additional hardware agents in the network and the sensing area can be divided into regions where each agent is responsible to localize the nodes in its predefined region. Each agent may send heartbeat signal periodically to check the situation of each other. For example, if the agent *A* does not receive a heartbeat signal from the agent *B* which is in its neighboring region, the agent *A* may take the responsibility of agent *B*.

4.2.3 Energy Harvesting Algorithms

Rahimi *et al.* [28] showed that when 40% of the total nodes are energy producing agents, then the lifetime of the network is guaranteed. Although this study is very important to show the effect of the energy producing mobile agents to the network lifetime, the 40% of value seems costly for sensor networks with thousands of nodes. Better techniques may be applied to reduce the percentage of energy-producing mobile hardware agents.

4.3 Sensor Nodes as Agents

4.3.1 Data Sampling Algorithms

Kho *et al.* [32] fixed the transmitting rate and data sampling schedule. These parameters may be variable in order to enable sensors to deliver data as soon as possible. This may predict the flooding with a smaller latency. Using the algorithm, time periods can be changed and adapted to different parts of environmental monitoring domain. Different WSN scenarios may also be simulated by changing topological configuration of the approach.

4.3.2 Communication Protocols

Kho *et al.* [36] experimented advantages of fixed and flexible routing. Coordination message packets and average computation time of a node for fixed routing are 100 times less than flexible routing which delivers

Table 4: Summary of agent technologies

Technology	Services	Drawbacks	Open research issues
Mobile software agents	Providing agent-based middleware support to WSN, using software agents in WSN, increasing network lifetime	Cost of middleware kernel to be installed on nodes, difficulty of a well-designed agent-based generic middleware for WSNs	Trade-off criteria for agent-based applications in WSN, transforming a usual WSN application to agent-based one, heuristics for optimization
Mobile hardware agents	Constructing network architecture, providing localization and energy harvesting	Economic cost of agent devices, single point of failure	Fault-tolerant and secure network architectures, more efficient energy harvesting
Nodes as agents	Data sampling, constructing network architecture, mission-task assignment	Redesigning agent-based algorithms on resource-constrained sensor networks	Energy efficient routing, path planning

more data to the base station. In flexible routing, only the nearest hop count is considered for energy yield. Nodes owing greater energy could be chosen on the path to the base station in order to construct energy-efficient paths.

In Strandens *et al.*'s approach [37] when the number of mobile sensors increases, the movement area of each node decreases thus, making it more difficult to move to necessary locations. An agreement approach between sensor nodes may be proposed to solve this problem, for example, a cluster-based movement procedure may be adapted.

Table 4 shows a summary of technologies with the services, drawbacks, and open research issues. Mobile software agents mainly provide the network with an agent middleware with the cost of installation of this middleware to all nodes in the network. Mobile hardware agents provide the network topology with the localization and energy harvesting services with the drawback of hardware costs. As the last technology, the sensor nodes are used as agents to support data sampling and task assignment services, wherein reconfiguration of the agent algorithms for sensors should be considered. The open research issues on these technologies may include software optimization, security, fault tolerancy, energy harvesting, efficient routing, and path planning.

5. Conclusions

In this paper, we initially provided a classification of agent technologies for sensor networks as mobile software agents, mobile hardware agents, and sensor nodes themselves as agents. Mobile software agent is a program including an executable task and a status data; and it has the ability to migrate from a sensor node to another one within the network. On the other side, a mobile hardware agent is a specialized powerful device which has the ability to traverse the network to collect information from sensor nodes. Moreover, each sensor node may be an agent itself, which leads to adaptive collaboration and coordination. We show that software agent technologies can be used as middleware platform support to sensor networks, and software agents can cooperatively help sensors accomplish the

applications of the whole network. Mobile hardware agents may execute complicated algorithms and they provide network architectures, localization, and energy harvesting. When a sensor node is modeled as agent, adaptive data sampling, flexible routing, and task assignment capabilities in the network are increased.

Secondly, we showed the objectives and pros and cons of each approach. The basic advantage of a mobile software agent is its decision on executing its code on one node or within the part of the network instead of that every node computing the tasks locally. Agent use may present the migration overhead; however, a well-designed agent-based middleware platform may provide basis for the good performance of WSN applications. The communication overhead of sensor nodes is reduced by mobile hardware agents, but the hardware agents have the maintenance costs. The algorithms which model the sensor nodes as agents aim to redesign the existing agent-based approaches by considering the needs of resource constraints of sensor nodes.

Lastly, we provided the open research issues for each agent technology class. We believe that our work may be a good starting point for any further studies in this topic.

References

1. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. "Wireless sensor networks: a survey", Elsevier Computer Networks, vol. 38, pp. 393-422, Winter 2002.
2. X. Bai, S. Li, and J. Xu. "Mobile Sensor Deployment Optimization for k-Coverage in Wireless Sensor Networks with a Limited Mobility Model", IETE Technical Review, vol. 27(2), pp. 124-37, 2010.
3. N. Uppu, B. Subrahmanyam, and R. Garimella. "An Energy Efficient Technique to Prolong Network Life Time of Ad Hoc Sensor Networks [ETPNL]", IETE Technical Review, vol. 25(4), pp. 154-60, 2008.
4. A. Alaybeyoglu, K. Erciyes, A. Kantarci, and O. Dagdeviren. "Tracking Fast Moving Targets in Wireless Sensor Networks", IETE Technical Review, vol. 27(1), pp. 46-53, 2010.
5. W. Gerhard. "Multiagent systems: A Modern Approach to Distributed Artificial Intelligence". The MIT Press, Cambridge, 1999.
6. A. Rogers, D. D. Corkill, and N. R. Jennings, "Agent technologies for sensor networks", IEEE Intelligent Systems, vol. 24 (2), pp. 13-7, Sep. 2009.
7. M. Chen, S. Gonzalez S, and V. C. M. Leung. "Applications and design issues for mobile agents in wireless sensor networks", IEEE

- Wireless Communications, vol. 14(6), pp. 20-6, Winter 2007.
8. V. Meritxell, A. R.A. Juan, and J. Cerquides. "A survey on sensor networks from a multi-agent perspective", In International Conference on Autonomous Agents and Multi-Agent Systems, Estoril, pp. 17-25, May. 2008.
 9. L. Tong, O. Zhao, and S. Adireddy. "Sensor networks with mobile agents", In IEEE International Symposium on Military Communications, Boston, pp. 688-93, Oct. 2003.
 10. E. Shakshuki, H. Malik, and M. K. Denko. "Software agent-based directed diffusion in wireless sensor networks", *Telecommun Syst.* vol. 38, pp. 161-74, Spring 2008.
 11. C. Fok, G. Roman, and C. Lu. "Agilla: A mobile agent middleware for self-adaptive wireless sensor networks", *ACM Transactions on Autonomous and Adaptive Systems*, vol. 4(3), Jul. 2009.
 12. Boulis, C. Han, and M. Srivastava. "Design and implementation of a framework for efficient and programmable sensor networks", In ACM International Conference on Mobile Systems, Applications and Services, San Francisco, pp. 187-200, May. 2003.
 13. P. Kang, C. Borcea, G. Xu, A. Saxena, U. Kremer, and L. Iftode. "Smart messages: a distributed computing platform for networks of embedded systems", *Oxford The Computer Journal, Special Issue on Mobile and Pervasive Computing*, vol. 47(4), pp. 475-94, Summer 2004.
 14. S. Gonzales, S. Vuong, and L. C. M. Leung. "A mobile code platform for distributed task control in wireless sensor networks", In ACM Workshop on Data Engineering for Wireless and Mobile Access, Chicago, pp. 83-6, Jun. 2006.
 15. E. P. Freitas, M. A. Wehrmeister, C. E. Pereira, A. M. Ferreira, and T. Larsson. "Multi-agents supporting reflection in a middleware for mission-driven heterogeneous sensor networks", In International Conference on Autonomous Agents and Multi-Agent Systems, Budapest, pp. 25-33, May. 2009.
 16. T. Harman, J. Padget, and M. Warnier. "A multi-layered semantics-ready sensor architecture", In International Conference on Autonomous Agents and Multi-Agent Systems, Budapest, pp. 5-9, May. 2009.
 17. P. Levis and D. M. Culler. "A tiny virtual machine for sensor networks", In International Conference on Architectural Support for Programming Languages and Operating Systems, San Jose, CA, USA, Oct. 2002.
 18. Available from: <http://www.tinyos.net/tinyos-1.x/doc/Xnp.pdf>. [last accessed on 2009 Nov 5].
 19. J. W. Hui and D. Culler. "The dynamic behavior of a data dissemination protocol for network programming at scale", In Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems, ACM Press, New York, pp. 81-94, 2004.
 20. T. Liu and M. I. Martonosi. "A middleware system for managing autonomic, parallel sensor systems", In ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming, 2003.
 21. M. Lotfinezhad and B. Liang. "Energy efficient clustering in sensor networks with mobile agents", In IEEE International Conference on Wireless Communications and Networking, New Orleans, pp. 13-7, Mar. 2005.
 22. R. C. Shah, S. Roy, S. Jain, and W. Brunette. "Data mules: modeling a three-tier architecture for sparse sensor networks", In IEEE Workshop on Sensor Network Protocols and Applications (SNPA'03), 2003.
 23. F. Diehl, J. Curtis, S. Rodriguez, A. Tosun, and D. Zhu. "Power-efficient real-time data collection using mobile robots", In The 28th IEEE Real-Time Systems Symposium (IEEE RTSS'07), 2007.
 24. B. McLaughlan and K. Akkaya. "Coverage-based clustering of wireless sensor and actor networks", In Proceedings of IEEE International Conference on Pervasive Services (ICPS'07), Istanbul, Turkey, 2007.
 25. T. Melodia, D. Pompili, V. C. Gungor, and I. F. Akyildiz. "A distributed coordination framework for wireless sensor and actor networks", In Proceedings of ACM Mobihoc, Urbana-Champaign, Illinois, USA, 2005.
 26. P.N. Pathirana, N. Bulusu, A. V. Savkin, and S. Jha. "Node localization using mobile robots in delay-tolerant sensor networks", *IEEE Transactions on Mobile Computing*, vol. 4(3), pp. 285-96, 2005.
 27. H. Lee, H. Dong, and H. Aghajan. "Robot-assisted localization techniques for wireless image sensor networks", In IEEE Conf. on Sensor, Mesh, and Ad Hoc Communications and Networks (SECON), Sept. 2006.
 28. M. Rahimi, H. Shah, G. S. Sukhatme, J. Heidemann, and D. Estrin. "Studying the feasibility of energy harvesting in a mobile sensor network", In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Taipei, Taiwan, Sep. 2003, pp. 19-24.
 29. T. A. Alhmiedat and S. H. Yang. "A survey: localization and tracking mobile targets through wireless sensors network", *PCNet*, 2007.
 30. H. Li, H. Miao, L. Liu, L. Li, and H. Zhang. "Energy conservation in wireless sensor networks and connectivity of graphs", *Theor. Comput. Sci.*, vol. 393, pp. 81-9, 2008.
 31. S. Sigurdsson, A. Lamarca, W. Brunette, D. Koizumi, and M. Lease. "Making sensor networks practical with robots", In First International Conference on Pervasive Computing, Lecture Notes in Computer Science, Springer-Verlag; 2002, 2414, pp. 152-66.
 32. J. Kho, A. Rogers, and N. R. Jennings. "Decentralised adaptive sampling of wireless sensor networks", In First International Workshop on Agent Technology for Sensor Networks, a workshop of the 6th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-07), Honolulu, Hawaii, USA, 2007.
 33. R. Willett, A. Martin, and R. Nowak. "Backcasting: adaptive sampling for sensor networks", In Proceedings of the 3rd International Symposium on Information Processing in Sensor Networks, California, pp. 124-33, 2004.
 34. G. Mainland, D. C. Parkes, and M. Welsh. "Decentralized, adaptive resource allocation for sensor networks", In Proceedings of the 2nd USENIX/ACM Symposium on Networked Systems Design and Implementation, Boston, pp. 315-28, May. 2005.
 35. P. Padhy, R. K. Dash, K. Martinez, and N. R. Jennings. "A utility-based sensing and communication model for a glacial sensor network", In Proceedings of the 5th International Joint Conference on Autonomous Agents and Multi-Agent Systems, Hakodate, Japan, pp. 1353-60, 2006.
 36. J. Kho, L. Tran-Thanh, A. Rogers, and N. R. Jennings. "Distributed Adaptive Sampling, Forwarding, and Routing Algorithms for Wireless Visual Sensor Networks", In 3rd International Workshop on Agent Technology for Sensor Networks, Budapest, Hungary, pp. 63-70, May. 2009.
 37. R. Stranders, A. Rogers, and N. R. Jennings. "A Decentralised, On-line Coordination Mechanism for Monitoring Spatial Phenomena with Mobile Sensor Networks", In 2nd International Workshop on Agent Technology for Sensor Networks, Estoril, Portugal, pp. 9-15, 2008.
 38. T. Le, T. J. Norman, and W. W. Vasconcelos. "Agent-based sensor-mission assignment for tasks sharing assets", In International Conference on Autonomous Agents and Multi-Agent Systems, Budapest, pp. 33-41, May. 2009.
 39. M. P. Wellman. "Market-oriented programming: some early lessons". Book chapter in S. H. Clearwater, *Market-based control: a paradigm for distributed resource allocation*. World Scientific Publishing Co., Inc., 1996.
 40. C. Y. Chong and S. P. Kumar. "Sensor networks: evolution, opportunities, and challenges", *Proceedings of the IEEE*, vol. 91(8), pp. 1247-56, Aug. 2003.
 41. D. D. Corkill, D. Holzhauser, and W. Koziarz. "Turn Off Your Radios! Environmental Monitoring Using Power-Constrained Sensor Agents", In First International Workshop on Agent Technology for Sensor Networks, Hawaii, pp. 31-8, May. 2007.
 42. Perrig, R. Szewczyk, J. D. Tygar, V. Wen, and D. E. Culler. "SPINS: security protocols for sensor networks", *Wireless Networks*, vol. 8(5), pp. 521-34, Sep. 2002.
 43. H. Karl and A. Willig. *Protocols and Architectures for Wireless Sensor Networks*. John Wiley and Sons Ltd., 2005.

AUTHORS



Orhan Dagdeviren received his BSc. and MSc. degrees in Computer Engineering from Izmir Institute of Technology. He is a PhD candidate in Computer Eng. at Ege University, working under the supervision of Professor Kayhan Erciyes. He is also a research assistant in Izmir Institute of Technology. His interests lie in the computer networking and distributed systems areas.

His recent focus is on graph theoretic middleware protocol design for wireless sensor and mobile ad hoc networks.

E-mail: orhandagdeviren@gmail.com



Ilker Korkmaz received the BSc. degree in Electrical and Electronics Engineering and MSc. degree in Computer Science from Ege University. He is a PhD candidate at International Computer Institute at Ege University, working under the supervision of Professor Dr. Mehmet Emin Dalkilic. He is also an instructor at the Department of Computer Engineering at Izmir

University of Economics. His research interests include computer networks and network security. His recent focus is on secure aggregation protocols in wireless sensor networks.

E-mail: ilker.korkmaz@ieu.edu.tr



Fatih Tekbacak received the BSc. and MSc. degrees in Computer Engineering from Izmir Institute of Technology. He is a PhD candidate in Computer Engineering at Ege University, working under the supervision of Professor Dr. Oguz Dikenelli. He is also

a research assistant in Izmir Institute of Technology. His interests lie in the software agents, access control of role based agents on multi agent systems and semantic web. His recent focus is about access control of dynamically changing ontologies on agent environments.

E-mail: fatihtekbacak@iyte.edu.tr



Kayhan Erciyes received a BSc. degree in Electrical Engineering and Electronics from the University of Manchester, MSc. degree in Electronic Control Engineering from the University of Salford and a Ph.D. degree in Computer Engineering from Ege (Aegean) University. He was a visiting scholar at

Edinburgh University Computer Science Dept. during his Ph.D. studies. Dr. Erciyes worked as visiting and tenure track faculty at Oregon State University, University of California Davis and California State University San Marcos, all in the U.S.A. He also worked in the research and development departments of Alcatel Turkey, Alcatel Portugal and Alcatel SEL of Germany. His research interests are broadly in parallel and distributed systems and computer networks. More precisely, he works on distributed algorithms for synchronization in mobile ad hoc networks, wireless sensor networks and the Grid. He is currently with the Izmir University Computer Engineering Department Izmir, Turkey.

E-mail: kayhan.erciyes@izmir.edu.tr