Journal of Computer and System Sciences 81 (2015) 487–495

Contents lists available at ScienceDirect

Journal of Computer and System Sciences

www.elsevier.com/locate/jcss

# Agent-based framework for sensor-to-sensor personalization

Nabil Sahli [a,*], Nafaa Jabeura [a], Mohamad Badra [b]

[a] German University of Technology, Muscat, Oman
[b] Zayed University, Abu Dhabi, United Arab Emirates

ARTICLE INFO

*Article history:*
Received 15 November 2013
Received in revised form 30 May 2014
Accepted 30 July 2014
Available online 13 November 2014

*Keywords:*
Sensor network
Personalization
Software agent
Mobile agent
Routing
Security
Energy consumption

ABSTRACT

In Wireless Sensor Networks, personalization has been seen by researchers as the process of tailoring services to fulfill requests of different users with different profiles. This vision ignores that individual sensors commonly have different profiles and contexts and therefore different needs. In this paper, we aim at extending personalization by allowing sensors to support each other with services that mutually fit their differences. To this end, we propose an agent-based framework where sensor nodes delegate software agents (static or mobile) to collect valuable data about the neighboring sensors and the spatial characteristics of their surrounding environments. We also show how this framework may be used to make the routing process more convenient for relying nodes in terms of energy consumption and security.



## 1. Introduction

Personalization has been addressed and implemented in a variety of fields. Technological advances have been very beneficial in getting closer to users, acquiring their explicit and implicit data, as well as acquiring relevant data on their surroundings. In this context, Wireless Sensor Networks (WSNs) can improve and expand the quality of services across a wide variety of settings. This is particularly possible thanks to the context awareness ability of sensors and their ability to adapt and support new events of interest. Several research works (see Section 2) have benefited from those capabilities to deliver personalized services to the end-user. In this paper, we argue that further benefits could be obtained, not only by delivering personalized services to the end-user, but also by adding personalization within the sensor network itself. This means that sensors should not only sense/process/forward/move according to their own capabilities and/or the end-user's preferences. They have also to maintain one-to-one relationships with their neighbors, by understanding their mutual needs. Sensors have thus to offer personalized services to their next hops to achieve sensor-to-sensor personalization. For instance, a sensor node S would only send to its neighbor R data in the format that R can process while making sure that R is trustworthy and has enough resources. This requires from sensor S to have an updated knowledge about its neighbors' status, capabilities, and context. This implies exchanging an important volume of messages that may not be supported by the available bandwidth and the current level of energy and processing capabilities of the sensors. It may also require collecting contextual data (e.g. characteristics of the space where these neighbors are operating) which are not necessary available at any of these neighbors. To this end, we believe that it is important to endow sensors with autonomy and intelligence

* Corresponding author.
*E-mail address:* nabil.sahli@gutech.edu.om (N. Sahli).

http://dx.doi.org/10.1016/j.jcss.2014.11.005
0022-0000/

allowing them to provide peers with the right data at the right time. Agent technology appears then as a serious candidate for this task.

An agent is a computer system which acts autonomously in its environment to meet its design objectives [1]. Thanks to their autonomy, agents can operate in an environment which is open, highly dynamic, uncertain, or complex [1]. Similarly, sensors are required to behave autonomously within a distributed network and adapt their behaviors to the changing environment without human intervention. In addition, sensors have to collect data about their neighbors (to provide them with personalized services) without compromising the overall performance of the network. In this context, the agent community has an adequate set of formalisms, algorithms, and methodologies which can address these challenges [2].

In the reminder of this paper, Section 2 explores the related works in personalization and agent use in WSN. Section 3 presents our agent-based framework which provides sensor-to-sensor personalization by allowing sensors to act autonomously and more intelligently. Section 4 outlines our proof of concept. Finally, Section 5 summarizes our contributions and future works.

## 2. Related works

Several research works, particularly in healthcare applications [3], smart-spaces [4], and mobile applications [5], have benefited from WSN capabilities to provide users with personalized services. For example, in order to achieve pervasive healthcare environments, sensor networks were used for a variety of purposes that range from simply setting an alarm volume according to the user's hearing abilities and the ambient noise level, to the complex tailoring of the user's entire eHealth environment [6].

Furthermore, thanks to the micro–nano technologies, the recent types of biomedical sensors are allowing personalization to be achieved more efficiently by maintaining and updating user's profile and data related to his/her context, general and specific preferences, physical and mental abilities, and other relevant parameters [6]. In smart environments, applications generally require situated, individualized, and personalized information to give optimal support to the user [7]. In addition to the information stored beforehand about the user, data on the current situation and user's activities are commonly acquired by on-body and off-body sensors. On-body sensors (e.g., biological signal sensors) are helpful to get the implicit feedback of users while off-body sensors have been used to acquire data on a variety of issues, including persons' identities, locations, gestures, focus of attention, and emotion.

Many other examples can be found in the literature. However, most of personalization efforts have been performed at the level of services delivered by the sensor network to the end-users. To the best of our knowledge, no research work has tackled the issue of personalization within the sensor network at the level of sensor-to-sensor communications. This could be explained by the fact that personalization has been always seen as an effort aiming to deliver services to a human being (as an end-user).

Regarding the use of software agents, many agent-based approaches have indeed been proposed to solve various problems in sensor networks [8]. More recently, powerful mote platforms have been developed by using intelligent Wireless Sensor Networks (iWSNs) [8]. As energy conservation is one of the main concerns in WSN, most agent-based approaches in WSN aim at enhancing the node life, in particular by introducing mobile agents. Indeed, mobile agents, when used in WSN, reduce the message traffic and thus save energy [8]. For example, in [9], mobile agents are used to reduce the communication cost by moving the processing function to the data rather than bringing the data to the sink. Each mobile agent has to carry a code to the source nodes and brings back aggregated data to the sink. In [10], the authors propose to reduce energy consumption of the WSN – to forecast water quality – by using data aggregation algorithms whereby mobile intelligent agents act as dynamic clustering points in the network. In [11], mobile agents store and gather metadata from nodes while minimizing route cost and maximizing battery level of sensors. The use of agents (particularly the mobile ones) does not only save energy. They may also allow a more efficient use of sensors' memories [8]. Indeed, since running all codes on a given node is often expensive and sometimes infeasible due to restrictions on local memory and processor, mobile agents can be deployed to support code distribution between sensors [12,13]. In terms of conceptualization, several research works have modeled sensor nodes as software agents (not mobile) to achieve adaptive data sampling (e.g., [14]), improve task assignment (e.g., [2]), and make data routing more efficient (e.g., [15]). In the next section, we propose a framework where each sensor can delegate some of its tasks to a mobile agent. This latter migrates to other nodes/platforms to collect relevant data which are needed to offer personalized services to next hops.

## 3. Agent-based framework

### 3.1. General concepts

Fig. 1 shows a layered framework which design philosophy has been inspired by the layered simulation model in [16]. The framework builds a parallel between a Real World (where the WSN is deployed to manage/monitor real resources) and a Virtual World (where software agents can behave/act on behalf of the real sensors).

In order to offer a personalized service to its neighbor, a sensor node has to take into account the environment context, the requirements and constraints of its neighbors, as well as its own goals and restrictions. To this end, and under such circumstances, the Real World environment is not necessary the best place for the following four reasons:

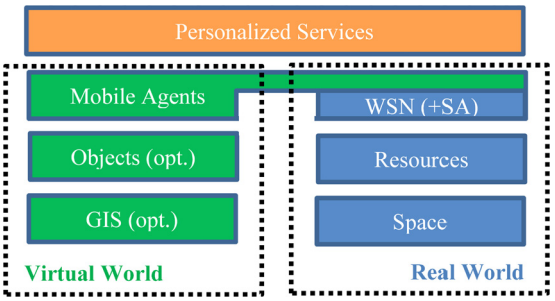
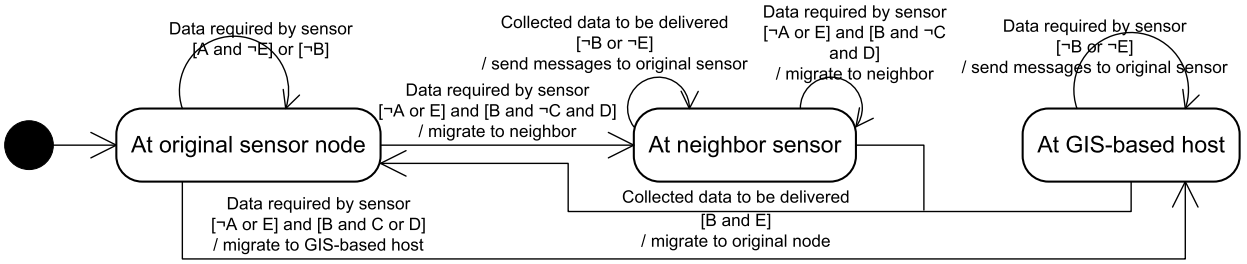



**Fig. 1.** General view of our agent-based framework.




**Legend**. A: feasible from original sensor; B: enough bandwidth for migration; C: many sensors needed for interaction; D: required data available at neighbor node; E: Interaction by message more expensive than migration; ¬: not.

**Fig. 2.** State diagram of Mobile Agent (MA).

- The process may involve the collaboration of several sensors with a high volume of exchanged messages. Communication is the main energy consumer function of a sensor.
- Offering a personalized service to the other nodes may require from a given sensor more processing. This is often very difficult if not unfeasible due to its limited memory and CPU.
- The decision/action of each sensor may depend on the spatial characteristics of the surrounding space (elevation, slope, etc.) and the location of the other resources/sensors. These data are, at best, partially available for a single node in the real world.
- Individual sensors have only a partial vision of the overall environment (status of the global network, context, etc.). As each sensor is only aware of its neighborhood, the undertaken actions are not necessary positive for the overall network.

To overcome the four problems depicted above, we endow each physical sensor S with two software agents: a Stationary Agent (SA) which resides in node S and a Mobile Agent (MA) which is initially at S but can migrate to a neighbor node or to a more enhanced Virtual World depending on the situation. We suppose here that each pair of SA and MA is exclusively attached to one single physical sensor. The state diagram of Fig. 2 depicts the different cases of MA migration. Basically, MA will leave its original node S only if personalizing a service at S would be expensive in terms of communication/energy or unfeasible for lack of data. MA may then migrate to one/few neighbor node(s) in order to collect/exchange relevant data. In some cases, when the process requires a big number of nodes and/or implies data which is not available within the sensors (e.g., spatial data), MA has to migrate to an enhanced Virtual World (mainly, a host based on a Geographic Information System (GIS)).

### *3.2. Virtual world*

The Virtual World is a platform where software agents can: (i) meet (i.e. exchange local messages) each other to share data (about their original sensor nodes) and (ii) optionally, access to the GIS data to apprehend the geographic characteristics of the space surrounding the current location of their original sensors (if such data is needed to offer a personalized service) as well as the resources to be managed/monitored (represented by stationary objects or agents [16]). Concretely, if MAs aim at meeting to exchange data between them, the Virtual World could be simply a super-node with extended memory and CPU capacities. However, if MAs need to access to the GIS data, the Virtual World would rather be a remote host (with extended processing and energy capacities) where a dedicated software platform provides MAs with spatial data (GIS database).

The MA, once its work is done within the remote node or platform, has to communicate with its original node S (more precisely with the SA) in order to feed it with the data required to offer a personalized service to S' neighbors. The MA has then to choose between migrating to S and sending a message to S. This choice depends on the network status and the volume of data to be sent (Fig. 2).

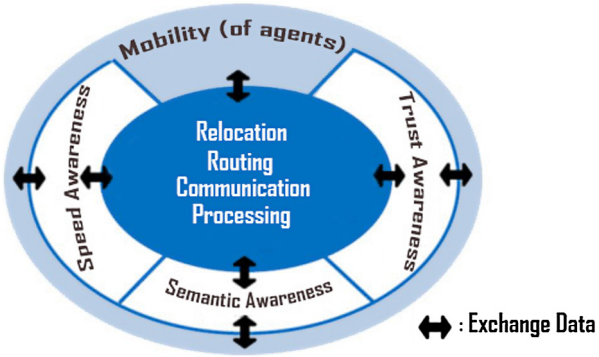



**Fig. 3.** Sensor endowed with extra capabilities.

The selection of the Virtual World and the association of each sensor to a particular Virtual World are not discussed in this paper. We mainly aim here at demonstrating the benefits of such meeting infrastructure. We can summarize the advantages as follows:

- Most of the inter-agent messages are local (in the Virtual World environment). Even a high traffic of exchanged messages will not really affect the performance of the WSN. Agent migration, if any, is the only noticeable overhead. However, a well-designed agent-based middleware platform may provide basis for the good performance of WSN applications [8]. We also estimate this overhead in Section 4.5 and demonstrate that it is much below the overhead generated by exchanging remote messages.
- Agents within the Virtual World can benefit from the extended capacities in terms of memory and CPU which are required to process data required later on for personalization. They can also get updated data about status/attributes of the many other sensors represented by their MAs.
- Each agent in the enhanced Virtual World (with GIS capabilities) has full access to all relevant data (needed to offer a better personalized service) including the spatial data of the surrounding space.

### *3.3. Personalization via sensor enhanced capabilities*

In order to enable sensors to provide personalized services to neighboring peers, sensors (and/or their agents) have to be endowed with extra capabilities, namely, semantic-awareness, trust-awareness, and space-awareness (see Fig. 3). Basically, sensor-based personalization may be carried out at three stages: data acquisition, data processing, and data communication. For instance, when a sensor, initially in an idle state, receives a request to acquire some data, it starts by assessing, if necessary, its own trust (using for example [17]) on the sender of the message. If this sender is trustworthy, the receiver starts by checking the data type requested. If this data type is not supported, the sensor notifies the sender that it is unable to acquire the requested data. Otherwise, the sensor identifies the needed data accuracy and sampling rate then collects that data. If data analysis is necessary, the sensor may carry out some processing, such as data filtering, data aggregating, and data formatting with respect to the expected level of details. Once the personalization of data processing is achieved, the sensor carries out the personalization of data communication by setting up the size of data packages while taking into account the QoS requirements of the beneficiary peer. In the three cases (data acquisition, processing, or communication), if the output is not personalized as expected, the receiver sensor may make some recommendations to the peer based upon its awareness about the current situation and its surroundings.

Fig. 3 summarizes the different capabilities of our sensors. Each sensor has the three common capabilities which are Processing, Routing, and Communication in addition to Relocation for mobile sensors. We propose to add four other capabilities to achieve personalization:

- Space-awareness: a sensor S needs to know the characteristics of the space in which it is operating in order to provide a better service to its peers. For example, in a highly dynamic environment where physical sensors are mobile, and realizing that it is not reachable by any other node, a node S may decide to move within the space while taking into account the spatial constraints (e.g., physical obstacles, terrain characteristics). Such space-awareness requires data which is provided by the Virtual World (equipped with GIS).
- Trust-awareness: a sensor may need to know if a neighbor node from/to which it gets/sends a message is trustworthy or not. This implies different processing (e.g., by encrypting data) or routing (e.g., by choosing another route) if the destination node is more or less trustworthy. Data about sensors trustworthiness may be collected at the sensor itself (via its SA) based on its own experience with the targeted node or at the Virtual World (via its MA) based on the feedback of the other nodes.
- Semantic-awareness: a sensor may perform a smarter forwarding if it can understand the semantic of the data. It will then avoid sending useless data to sensors. More details are in [18].

- Mobility (of agent): this capability is supported by the MA and aims at collecting the necessary data from the Virtual World to feed all the other capabilities (as shown in Fig. 3).

### 3.4. Discussing enhanced capabilities

Endowing sensors with the enhanced capabilities mentioned above requires discussing their impact on two main sensor constraints, namely, computing complexity and energy usage. In particular, agents, when embedded in devices such as sensors, must be circumspect in their use of energy since these nodes are often very constrained by nature [19]. It is here worth mentioning that mobile agents, when used in WSN, reduce the message traffic and thus save energy [8]. Several research works (see Related Works) have indeed confirmed this. In addition, in our framework, agents migrate only if this does not affect the bandwidth (see Fig. 2). Second, the Space-awareness capability implies being in a Virtual World (with GIS data) where resources (computation and energy) are relatively abundant. Finally, making sensors aware of the trustworthiness of their neighbors and the semantic of the forwarded data requires from each sensor to collect data (from its neighborhood) and performs extra processing. Nevertheless, when these tasks are performed by the MA in the Virtual World (e.g., a super-node with extended memory and CPU capacities), the WSN overall performance is not really affected. The network may provide better services (by supporting trust and semantic) at a cost of installing more super-nodes (if necessary), allowing agents to migrate through the network (when possible), and adding little complexity on super-nodes (where resources are not as critical as in simple sensor nodes). The only case where simple sensor nodes may have to perform significant extra processing are when MAs cannot migrate (e.g., due to limited bandwidth) to the Virtual World. In this situation, a sensor which is running out of resources may suspend its personalization activities and focus on its primary functions (data acquisition, processing, and routing). To conclude here, sensors endowed with enhanced capabilities offer better services to each other and thus to the end-users. However, this implies more or less extra usage of resources (energy and processing). We think that each WSN designer, depending on the constraints and objectives, has to find a compromise between quality of service (provided by the enhanced sensor capabilities) and resource usage.

### 3.5. Real-life application

The framework presented in this paper is mainly relevant for applications which involve large-scaled and highly dynamic environments such as areas affected by natural disasters. WSNs have been used to manage and monitor large areas during natural disasters such as forest fires. Nevertheless, sensors when deployed in such harsh environments are operating under more challenging constraints: unreliable communication bandwidth, untrusted nodes (as many sensors will be affected by the high temperatures), high flow of sensed data, and space (geographic) dependency (since the geographic characteristics of the terrain highly influence the wildfire spreading and its management). Our framework deals with all these issues by providing a meeting infrastructure where spatial information is accurate, communication is at lower cost, and data about nodes is available for all peers. More details about the deployment of such solution for the forest fire problem can be found in [20].

## 4. Proof of concept on security-based routing

Most of existing routing protocols are based on one main criterion – power consumption – at the expense of other aspects such as reliability, security, or efficiency [21]. To achieve a more sensor-to-sensor personalized routing, many requirements have to be taken into account apart from power saving, namely, sensor mobility, location, space, semantics, quality of service, and trust/security [21]. In [21], we already proposed a generic multi-criteria routing framework where the selection of the best next neighboring hop is personalized according to the environment (context), the end-user preferences, forwarder's requirements and constraints, and receiver's constraints and requirements (sensor-to-sensor personalization). As suggested by our framework in [21], the sender node S chooses the next hop based on the data collected about its neighbors (level of energy, location, supported security level, semantics, capabilities, etc.). However, collecting such information from neighbors requires sending/receiving a large number of messages, which may have a big impact on the energy consumption of the sensor. Our agent-based framework solves this problem and at the same time provides sensors with more capabilities.

In this section, and for the sake of simplicity, we focus on the security criterion and show how sensors can take advantage from our framework to collect/exchange security-related data within the network. This data will ultimately allow sensors to enforce personalized routing by supporting mutual preferences related to supported security protocols and encryption mechanisms.

### 4.1. Security requirements

In order to achieve the objectives mentioned in the previous paragraph, sensors first need to know which security mechanisms are supported by their neighbors. When a node N sends for instance a confidential message through the network, it has to make sure that the receiver (next hop) has adequate security mechanisms to maintain the same security level. This requires that each sensor of the network exchanges the list of supported security mechanisms with its neighbors on beforehand. Moreover, if a group of sensors, that we call here circle of trust, decide to send secret data between each other,

they have to share a secret key. To this end, they have to exchange some data to generate the key and then periodically communicate to update this key. If the circle of trust is formed by few sensors or if these sensors are 1-hop distance from each other, the exchange of messages needed to share the key should not really affect the WSN traffic or their respective energy level. However, in the opposite cases, these messages may result in a high communication overhead and thus speeding up the depletion of the limited energy of sensors. In such cases, the sensors belonging to the circle of trust might not be able to establish/update the shared key possibly due to communication failure.

Furthermore, certificate or secret key-based authentication may not be sufficient to build a trustworthy path between two (or more) nodes. In fact, nodes may have disruptive misbehaviors even though they were successfully authenticated. Selecting the most trustworthy next hop is, thus, more appropriate than making a random choice between all available authenticated nearby peers. To this end, every sensor needs to maintain information on the trust that it has on its neighbors and use this information when transmitting data to a given destination through a set of intermediate nodes. The paper's aim is not to determine a way to calculate trust. We only suppose that each node N maintains a trust factor on each neighbor S (or other remote nodes which are of interest to N) based on N's past experience with S and the reputation of S within the network. Each trust factor may be updated periodically to reflect the changing behaviors of different nodes. Establishing such trust relationships between nodes requires exchanging a large volume of messages between nodes. It also requires extra processing and memory capabilities in order to run the necessary trust and reputation mechanisms. These requirements will definitively affect the energy consumption of the sensor network. Maintaining the trust values requires storage capabilities as well. However, since a sensor has only to store a few values (a few bytes) for a restricted number of nodes (neighbors), the default storage space of most current sensors is more than enough.

### *4.2. Technical requirements and assumptions*

To address the constraints mentioned in the previous sub-section, we use our agent-based framework where software agents collect/exchange data from/between each other on behalf of their original sensors. More specifically, each sensor node delegates a mobile agent MA (if the conditions are met, see Fig. 2.) which may migrate to a neighbor, a super-node, or to a GIS-enhanced Virtual World depending on the environment constraints (discussed previously in Fig. 2). Practically, each node has to run a mobile agent platform for WSN. Examples of such platforms include Agilla, MAPS, and AFME. A comparison between the features and capabilities of these platforms can be found in [22]. Such a platform can create an MA, optimize it (in terms of code and state size), encrypt it, serialize it into a message, and then send it to the Virtual World. For our "Security" scenario, we suppose that the Virtual World is a super-node (not necessarily hosting a GIS database). We also suppose that a group of sensors (typically neighbors) agree on beforehand to virtually meet (via their MAs) at a specific super-node. This supposes that super-nodes are already known by all nodes. The agreement between sensors can be reached by exchanging messages and may be done once for static WSNs. This agreement also means that each MA, before migrating, knows with which sensors' agents it has to interact once it reaches the super-node. For example, if sensors S1, S2, and S3 agree to meet at a specific super-node, MA1 (the agent of S1) knows then that it has to find the agents of S2 and S3 once it arrives at the super-node.

Regarding the super-node, it also has to run the same mobile agent platform as the one used by the nodes or a compatible one. We also suppose that the super-node has a MODerator agent (MOD) which is in charge of moderating all exchange sessions between the hosted MAs. These MAs are complying with the same communication languages (e.g., KQML or FIPA-ACL) and protocols within the host.

### *4.3. Super node functioning*

Upon arrival to the host, a mobile agent MA is immediately received and deserialized by the host platform (most existing platforms are capable of automatically detecting the presence of the coming agent once they arrive to the host). During its stay at the super-node, MA has to obey the host functioning rules described below.

Signing up: MA has to sign up by specifying its full identity (including its own identity and the identity of its corresponding sensor) as well as the identities of the agents of the sensors with which it wants to exchange data. These latter identities are known on beforehand (as we discussed earlier). MA may also indicate to the moderator MOD if it requires all the indicated agents to be present before staring exchanging messages with them or not. The MOD may refuse the MA's request to sign up for different reasons, including super-node already at its maximum capacity and suspicious MA.

Notification: After signing up, the MOD notifies MA with the identifier of each available agent that MA requested to talk to. If MA had specified during the signing up phase that it requires all agents to be present to start the session, the MOD will only notify MA when all these agents are present.

Exchange session: MA may start sending/receiving messages to/from the other agents. For instance, they can share a secret key, update a secret key, exchange/update their table of trust factors, exchange list of supported security mechanisms, etc. During the session, MA may communicate with its sensor via messages if it has to report important data or decisions. It can even migrate back to its sensor node if necessary as we already mentioned in Fig. 2. Finally, and during the session, the MOD can notify the group if a late node has signed up or if a member of a group has signed out.

Signing out: When MA finishes its session and if it opts for migrating back, it has to sign out by sending a *sign_out* notification to the MOD so that the latter knows which sensors are still there at each time. The MOD may also periodically

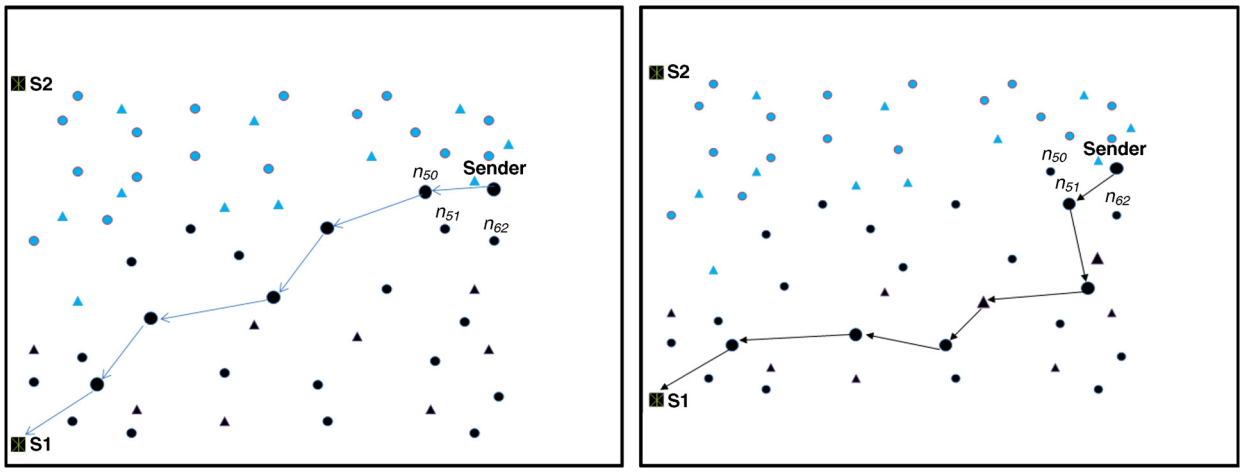



**Fig. 4.** (a) Shortest routing path. (b) Most secure routing path.

check which agents are alive by sending them a ping message. This will allow the MOD to detect agents which were killed (process has been ended) without signing out. The MOD may also exclude a given MA from the host for diverse reasons, such as misbehavior or being inactive agent for a long period of time.

### *4.4. Prototype*

Our Virtual World is implemented as a java application. The sensor-like behavior (e.g., routing algorithms) of agents within the super-node is ensured by the WSN simulator engine Castalia which is based on OMNeT++ [23]. Our simulations were carried out on randomly distributed sensors of different types (different graphic representations on Fig. 4(a)).

Each mobile agent MA, corresponding to a given sensor node N, and after exchanging data with its peers in the Virtual World, sets up a trust and security factor for each of these peers. These values are then sent to the sensor node N. In the real world, and when the node N receives a packet (to be routed until a final destination S1), it selects the most appropriate next hop based on the security and trust factors [21] which were previously collected and processed by the MA. This illustrates the sensor-to-sensor personalization facet.

Our experiments show that the resulting routing paths (e.g., Fig. 4(b)) may be sometimes longer than those found by traditional routing algorithms (e.g., Fig. 4(a)). However, they support in a better way the individual needs of sensors along the communication pathways while reducing the risk of communication failure, for example by avoiding untrusted peers. The prototype can also combine different selection criteria apart from security and trust (not presented in this paper). The node which will be selected as the next hop is the one which overall score (combination of all criteria' scores) is the highest. Moreover, depending on the end-user's requirements, node N may consider different weights of the selection criteria applied to each neighbor. For example, if the end-user indicates security as the main constraint, security criteria will have the biggest weight in the formula [21] to select the most appropriate next hop. This is the end-user personalization facet. Practically, the sensor node N receives all values corresponding to all selection criteria for all neighboring peers. Sensor N applies then a weighted formula which uses all the values processed by the MA to select the best next hop. Finally, the combination of the two personalization facets described here determines the final routing path.

### *4.5. Overhead evaluation*

As previously mentioned, it is important to estimate the overhead generated by agent migration and check if the energy consumption of individual nodes (and the entire network) is not affected. To do so, we propose to estimate the energy consumption for a small scenario involving nodes which want to exchange some data between each other. We compare the energy consumption of the same scenario but with two different approaches: exchanging remote messages between sensors and migrating mobile agents MAs to a super-node where they can exchange data.

Since the network overhead associated with agent migrations is directly related to the MA size, both the agent code and its state should be optimized [24] even though agent code size is typically far larger than its corresponding state size [25]. For example, the source code of the MA can be optimized so that the size is reduced by half [24] and can later be compressed to be even lighter. Moreover, in many cases, the transfer of code is unnecessary since the mobile agent can load the code once it arrives to the host [25]. A sensor node can thus send only the state information of the MA. Consequently, an MA may end up with a few hundreds bytes size.

In order to build the energy model for our scenario, we make the assumption that the data message (and not the signal message) sent by a sensor node to its next selected hop has a relatively similar size (around 250 bytes) than an agent MA. This assumption is plausible according to [25,26]. A sensor which transmits a $k$-bit message to another sensor located at

distance $d$ using the radio model of [26] will consume the following energy: $E_{elec} * k + \epsilon_{amp} * k * d^2$ where $E_{elec}$ and $\epsilon_{amp}$ are parameters related to the sensor radio. Using the assumptions of [26], $E_{elec} = 50$ nJ/bit and $\epsilon_{amp} = 0.1$ nJ/bit/m$^2$. According to these formulas, it is clear that the dissipated energy depends on the size of the message to be sent as well as the distance to the next hop. It is worth mentioning here that the distance has high impact on the energy consumption (because of the $d^2$ term in the formula) than the size of the message. This is one of the reasons why our agent-based framework is aiming at reducing the number of messages exchanged between remote sensors.

A sensor now which receives a $k$-bit message from another sensor using the same radio model will only consume: $E_{elec} * k$, which is a linear function depending on the size of the message and which does not depend on the distance.

Let us suppose that we have a group of $N$ sensors which want to exchange data about their supported security mechanisms. Each node has to send a minimum of $N-1$ messages. It should also receive a minimum of $N-1$ message (one from each node). If we suppose (for the sake of simplicity) that all messages are of equal sizes and that all $N-1$ nodes are at equal distance from S, the total energy dissipated by node S will be then: $(N-1) * (2 * E_{elec} * k_{message} + \epsilon_{amp} * k_{message} * d^2)$ (by calculating both sent and received messages). On the other hand, if sensor S sends its MA to a super-node in order to exchange the same data with the other $N-1$ nodes, the total dissipated energy will be: $2 * E_{elec} * k_{MA} + \epsilon_{amp} * k_{MA} * d^2$.

Given that the size of the optimized mobile agent is close to the size of a message, and with only $N = 3$ (the group is composed of only 3 nodes), sending an MA will be less expensive for S than exchanging one message with each of the other 2 nodes (we suppose here that the super-node hosting the MAs is at a distance $d$ from S). The advantage of MAs is even higher if $N$ is becoming larger or if S has to exchange more than one message with each node.

In our agent-based framework, the entire load goes to the super-node. We thus compare the overhead cost on this super-node when hosting MAs sessions. To do so, we suppose that sensor S has to send/receive two messages to/from each node of the group. If S communicates remotely (without agents), the total dissipated energy will be: $2 * (N-1) * (2 * E_{elec} * k_{message} + \epsilon_{amp} * k_{message} * d^2)$.

The super-node will receive $N$ agents and will have to send them back to their nodes (in worst case). The total dissipated energy will be: $N * (2 * E_{elec} * k_{MA} + \epsilon_{amp} * k_{MA} * d^2)$. For the same reasons, this energy consumption is close to what a single sensor would have spent if it communicates remotely. Increasing the number of messages sent by each sensor will tilt the balance even more in favor of the super-node. In addition, the total energy dissipated by the $N$ nodes will be $2 * N * (N-1) * (2 * E_{elec} * k_{message} + \epsilon_{amp} * k_{message} * d^2)$ without agents and $2 * N * (2 * E_{elec} * k_{MA} + \epsilon_{amp} * k_{MA} * d^2)$ by using MAs and super-node. Here again, our agent-based approach is winning.

Nevertheless, the calculations above are supposing that a super-node is at a reasonable distance $d$ from all nodes. This may not be always possible. In addition, if the size of MA is larger than the message (e.g., 3 to 5 times more) and $N$ (number of nodes) is low (2 or 3), it may not be more beneficial to send the MAs to a super-node in terms of energy consumption. Ideally, the stationary agent (SA) (see Section 3.1) will evaluate whether the MA has to migrate to the super-node or not, depending on the distance to the super-node, the number of nodes with which the sensor wants to talk with, and the number of messages to be exchanged. If SA finds out that it is better to send MA, the latter will not migrate before all the other concerned sensors decide to also send their respective MAs. We purposely omit details regarding this issue from the present paper.

## 5. Conclusion and future works

In this paper, we proposed an agent-based framework for WSN where sensor nodes delegate software agents (static or mobile) to collect valuable data about the neighbouring sensors and the surrounding environment. It also gives accessibility to more refined data (GIS). All this data can then be used by sensor nodes to provide personalized services to each other. Our framework also shows the complementarity between physical sensors and software agents: while sensors are getting data from the field (Real World), agents are collecting data from the Virtual World. Putting both types of data together makes sensor nodes more intelligent and more autonomous.

We have already showed, in previous research, how sensor-to-sensor personalization can be achieved for specific tasks such as routing [21] and relocation [20]. The present work actually gives the supporting framework for these applications. We are currently working on defining other aspects and applications for the sensor-to-sensor personalization. We are particularly interested in endowing sensors with a stronger trust and reputation model so that they can provide more secure services to their neighbours, with more semantic-awareness to avoid forwarding useless data to the other sensors, and with more space awareness to provide services which can take the geographic characteristics into account.

Regarding the agent technology, we are working on building our own platform which will be able to provide agents with mobility, space awareness, and efficiency. We are indeed confident about the importance of this technology to solve many WSN issues. Indeed, even experimental sensor agent technology has become sufficiently reliable for operational use in the field. We do believe that the permanent deployment of sensor–agent networks is close.

## Acknowledgments

This research has received Research Project Grant Funding from The Research Council of the Sultanate of Oman Research Grant Agreement No. (ORG DU ICT 10 003).

## References


[1] W. Gerhard, Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence, MIT Press, Cambridge, 1999.
[2] A. Rogers, D. Corkill, N. Jennings, Agent technologies for sensor networks, IEEE Intell. Syst. 24 (2) (2009) 13–17.
[3] F. Miao, X. Miao, W. Shangguan, Y. Li, MobiHealthcare System: body sensor network based M-health system for healthcare application, E-Health Telecommun. Syst. Netw. 1 (2012) 12–18.
[4] J. Byun, B. Jeon, J. Noh, Y. Kim, S. Park, An intelligent self-adjusting sensor for smart home services based on ZigBee communications, IEEE Trans. Consum. Electron. 58 (3) (2012) 794–802.
[5] C. Chang, S. Hsu, Personalized service provision in a context-aware shopping environment, in: Fourth International Conference on Computational Intelligence, Communication Systems and Networks, CICSyN, 2012, pp. 103–108.
[6] P. Pharow, B. Blobel, P. Ruotsalainen, F. Petersen, A. Hovsto, Portable devices, sensors and networks: wireless personalized eHealth services, in: Proceedings of MIE'2009, 2009, pp. 1012–1016.
[7] S. Hämmerle, M. Wimmer, B. Radig, M. Beetz, Sensor-based situated, individualized, and personalized interaction in smart environments, in: Proceedings of GI Jahrestagung, 2005, pp. 261–265.
[8] O. Dagdeviren, I. Korkmaz, F. Tekbacak, K. Erciyes, A survey of agent technologies for wireless sensor networks, IETE Tech. Rev. (28) (2011) 168.
[9] H. Malik, E. Shakshuki, Data dissemination in wireless sensor networks using software agents, in: HPCS 2007, IEEE Computer Society, 2007, p. 28.
[10] M.S. Garcia, D. Carvalho, O. Zlydareva, C. Muldoon, B.F. Masterson, M.J. O'Grady, W.G. Meijer, J.J. O'Sullivan, G.M.P. O'Hare, An agent-based WSN for water quality data collection, in: UCAmI, in: LNCS, vol. 7656, Springer, 2012, pp. 454–461.
[11] Liu Tang, Feng-Cai Fang, Mobile agent based metadata framework for heterogeneous wireless sensor network, in: International Conference on Educational and Information Technology, vol. 1, ICEIT, 17–19 Sept., 2010, pp. V1-446–V1-449.
[12] Available from http://www.tinyos.net/tinyos-1.x/doc/Xnp.pdf, last accessed on 2013 January 30.
[13] J.W. Hui, 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, 2004, pp. 81–94.
[14] J. Kho, A. Rogers, 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, Hawai'i, USA, 2007.
[15] J. Kho, L. Tran-Thanh, A. Rogers, N.R. Jennings, Distributed adaptive sampling, forwarding, and routing algorithms for wireless visual sensor networks, in: 3rd Int. Workshop on Agent Technology for Sensor Networks, Budapest, May, 2009, pp. 63–70.
[16] N. Sahli, B. Moulin, EKEMAS, an agent-based geo-simulation framework to support continual planning in the real-word, Appl. Intell. 31 (2) (2009) 188–209.
[17] H. Chen, H. Wu, X. Zhou, C. Gao, Agent-based trust model in wireless sensor networks, in: 8th ACIS Int. Conf. on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, vol. 3, 2007, pp. 119–124.
[18] N. Jabeur, Y. Iraqi, N. Sahli, Semantic routing in wireless sensor networks: challenges and opportunities, in: IEEE Global Information Infrastructure Symposium, Tunisia, June, 2009.
[19] S. Shen, M.J. O'Hare, G.M.P. O'Grady, Fuzzy-set-based decision making through energy-aware and utility agents within wireless sensor networks, Artif. Intell. Rev. 27 (2007) 165–187.
[20] N. Sahli, N. Jabeur, Agent-based approach to plan sensors relocation in a virtual geographic environment, in: Proceedings of NTMS 2011, 7–10 February, Paris, France, 2011.
[21] N. Sahli, N. Jabeur, I. Khan, M. Badra, Towards a generic framework for wireless sensor network multi-criteria routing, in: Proc. of 4th Int. Workshop on Wireless Sensor Network: Applications, Developments, and Trends. In conjunction with the 5th IEEE International Conference on New Technologies, Mobility, and Security (NTMS'2012, 1–6), May 7–10, 2012, Istanbul, Turkey, 2012.
[22] F. Aiello, A. Carbone, G. Fortino, S. Galzarano, Java-based mobile agent platforms for wireless sensor networks, in: Proceedings of the 2010 International Multiconference on Computer Science and Information Technology, IMCSIT, 18–20 Oct., 2010, pp. 165–172.
[23] Omnet website, http://omnetpp.org/component/content/article/9-software/3478.
[24] D. Gavalas, An experimental approach for optimising mobile agent migrations, The Computing Research Repository, December 2012.
[25] G.P. Picco, Mobile agents: an introduction, Microprocess. Microsyst. 25 (2) (2001) 65–74.
[26] W.R. Heinzelman, A. Chandrakasan, H. Balakrishnan, Energy-efficient communication protocol for wireless microsensor networks, in: Proceedings of the 33rd Annual Hawaii International Conference on System Sciences, vol. 2, 4–7 Jan., 2000, p. 10.