A Novel Power-Efficient Middleware Scheme for Sensor Grid Applications

Nikolaos I. Miridakis, Vasileios Giotsas, Dimitrios D. Vergados, and Christos Douligeris

Department of Informatics University of Piraeus 80, Karaoli & Dimitriou St., GR-185 34 Piraeus, Greece
{nikozm,vergados,cdoulig}@unipi.gr, v.giotsas@ieee.org

Abstract. Sensor grid deployments integrate wireless sensor networks (WSNs) and Grid Computing (GC) into a merged platform. A middleware architecture is a prerequisite for sensor grids in order to bridge the two heterogeneous technologies and efficiently support aggregated grid services available to a large number of grid users. On the other hand, the energy conservation of the participating sensor nodes is an essential factor for QoS provisioning, thereby extending WSNs survivability and providing diversity to potential grid services. For the best of our knowledge, power awareness for middleware architectures for sensor grids has never been studied in the literature so far. The rationale of our work employs a scheduler which provides QoS to the grid users from an energy awareness perspective by interacting with an appropriate resource manager. Our simulations show the effectiveness of the proposed scheme whereas a proxy-based middleware for sensor grids has been adapted.

Keywords: Sensor Grids, Power Efficiency, QoS scheduling.

1 Introduction

Wireless sensor networks (WSNs) are one of the most rapidly evolving research and development fields for microelectronics. Their applications are countless, and the market potential is huge. Recent advances in micro-electromechanical systems (MEMS) have led to the creation of small sensor nodes which integrate several kinds of sensor components such as a central processing unit, memory and a wireless transceiver [1, 2]. These sensor components have been characterized as low-cost, low-power and self-contained instruments with limited sensing, data processing, and wireless communication capabilities. The most important applications of WSNs include environmental and habitat monitoring, healthcare monitoring of patients, weather monitoring and forecasting, military and homeland security surveillance, tracking of goods and manufacturing processes and safety monitoring of physical structures and construction sites, smart homes and offices [3].

Nevertheless, sensor nodes still remain resource constrained due to their limited bandwidth range and computation capabilities. Thankfully, WSNs consist of hundreds (sometimes thousands) sensor nodes deployed and aggregated over a certain wide

N. Bartolini et al. (Eds.): QShine/AAA-IDEA 2009, LNICST 22, pp. 476–492, 2009.

area, so the computational burden is therefore distributed among the nodes. Thus, WSNs are important distributed computing resources that can be shared by different users and applications [4].

Grid Computing provides a federation of heterogeneous computational servers and collaborating systems which are communicating through high-speed network connections. Many industries have recognised the importance of grid computing for 'e-science' where the grid has been employed extensively in the fields of bioinformatics, engineering design, business, manufacturing, environmental control and weather forecasting [1, 5, 6].

The combination of WSNs and grid computing under a sensor grid architecture (*sensor grid* in short) takes advantage of all the strengths and benefits of sensor networks and grid computing resulting in a single integrated platform [1, 7]. Thus, a sensor grid may combine real-time data about a wide unit area with vast computational resources derived from the grid architecture. A typical sensor grid framework is shown in Figure 1a. There is a trade off thought between the two merged technologies. On one hand, sensor nodes have constrained resources as they monitor the environment on a real-time basis while the resource-full grid infrastructure promises solutions to computational and communication tasks according to the ever-increasing needs of users.

There are mainly two approaches on a sensor grid deployment; the centralized and the distributed approach. In the centralized, sensor nodes and sensor networks are connected directly to the grid. High-speed communication links are necessary for this approach where all computational tasks take place on the grid. The main drawback of this approach is the fact that it leads to excessive communications among the nodes which rapidly depletes the batteries resulting to network partitioning, a rather undesirable choice. Also, possible communication failures in some nodes, such as bad radio propagation conditions, jamming and interference, could result in a general breakdown of the system. The distributed approach is more robust and efficient technique since it allows all computational and decision making jobs to be performed within the sensor network according to their resources and capabilities at a real-time basis [1].

Sensor grids being a relatively new area of research, there are many issues left unaddressed regarding their design. Moreover, because WSNs are usually based on proprietary designs and protocols, it is a challenging task to integrate them with the standard grid architecture and protocols [3]. In this paper, we analyze the issues and challenges present in the integration of WSNs and the Grid considering a distributed approach managed locally in each sensor area (a sensor area consists of a WSN or a cluster of several WSNs connected with the grid via virtual organizations). We also describe a proxy-based architecture, a middleware component enhancing the effectiveness of the overall framework, as shown in Figure 1b.

The middleware plays the role of an appropriate interface which is capable of providing functionalities such as normalizing and synchronizing the communication between sensor nodes and the grid. Furthermore, the proxy middleware model takes into consideration the limited power resources of sensor nodes making it an energy-aware architecture which has as its main scope tha preservation of the resources of

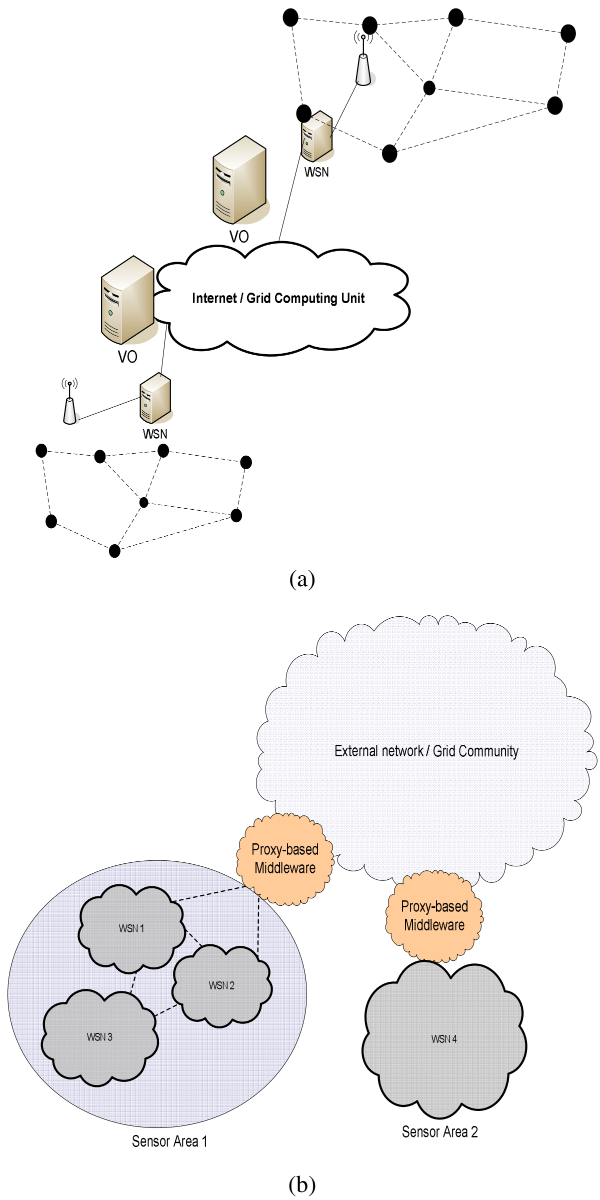

Fig. 1. (a) Sensor Grid framework. (b) The middleware connecting platform.

individual wireless nodes. As many symbols are used in this paper, Table 1 summarizes the most important ones.

Table 1. Summary of important symbols used

Symbol	Definition
$d_{i,j}$	Distance between node i to j
d_{eff}	Maximum effective transmission range
V_g	User group credentials
V_u	User profile
C	Service Class
N_S^c	Number of services for service class C
$\hat{p}^j$	Power consumption of node j for a single service
P_c^j	Total power consumption of node j for service class C
$Sh\_P_x[]$	An array which holds all shortest paths within subset X
m	Number of hops from a requested node to the gateway
ξ	Number of all paths from a node to the gateway
$sig\_P_c^j$	Signaling cost from the gateway to node j

The rest of this paper is organized as follows. In section 2, we discuss the most important compatibility problems that the two technologies encounter in order to provide an integrated platform. In section 3, the proposed model is presented in detail. Performance evaluation results are given in Section 4, followed by concluding remarks in Section 5.

2 Design Issues and Challenges

In this section we discuss the most important differences of the two merged technologies within sensor grid architecture. A natural approach to integrate sensor nodes into the grid is to adopt the grid standards and APIs. The Open Grid Services Infrastructure (OGSI) [14] establishes web services based on XML, SOAP and WSDL formats. However, since sensor nodes have limited resources and computational capabilities, as mentioned earlier, it may not be feasible to manipulate sensor data and to encode them into SOAP envelops using XML formats. Therefore, many grid services may be too complicated for the capabilities of the common wireless sensor nodes [6].

Moreover, most grid processes and applications use existing internet protocols to exchange messages, e.g. TCP, FTP, HTTP. Sensor networks, on the other hand, make use of low-level protocols (energy efficient MAC and routing protocols) due to their nature [2, 13]. Hence, the direct communication of a WSN with the grid is not feasible without appropriate interface.

Power management is one of the major issues in WSNs and sensor grid deployments. The grid infrastructure must be aware of the power/energy status of the nodes of each sensor area in order to make the most efficient and robust decisions since the availability of a WSN depends not only on its average load but also on its power resource constrains [3, 8].

An appropriate scheduler is definitely one of the most important and most complex operations in a proxy-based middleware deployment for sensor grids. The role of a scheduler in a typical WSN is to achieve load balancing and to avoid energy dissipation among the nodes, thus extending the network's lifetime and preventing network partitioning. In a sensor grid infrastructure, sensor nodes might be consisting of several wireless sensors, each used for a different purpose (temperature, sound, light, vibration etc) [3, 5]. Each type of data is collected by different types of wireless transceivers placed on the same device. A scheduler (combined with an appropriate resource manager, as explained in the next section) should control the on/off mode of the transceivers of every sensor device within the sensor area according to the needs of grid users as well as the available sensor resources. Furthermore, quality of service (QoS) is one very important issue in sensor grid networks. QoS is associated with the personalization logic that defines the service class for each user or user group responding to the grid. The personalization logic within the grid infrastructure branches the available services into classes according to the type of user, e.g. administrator, unsubscribed user, academic staff, commercial user, government. QoS factor is performed along with personalization logic coherently. Thus, an efficient scheduler should take into account the personalization logic, the QoS and the resource constraints of the nodes in a sensor area in order to provide suitable services both for the WSN and the grid.

3 Description of the Proxy-Based Middleware

In this section we describe the proposed middleware framework and we analyze its components and their functionalities in detail. The diagram in Figure 2 shows the deployment of the proposed proxy-based middleware infrastructure [9].

3.1 System Overview

The gateway is a station which collects the information from all the sensor nodes within its sensor area through the reception of wireless MAC frames transmitted directly from the WSNs. It also holds a record file with all the sensor ids which participate in the communication process (to be discussed subsequently). Afterwards, the gateway sends the aggregated row data flows to the middleware via a wired link. First, the preprocessing component evaluates the received data by purifying them from possible anomalies and aberrations due to propagation attenuation and atmospheric interference that the transferring process might cause. It also isolates individual MAC frames from the consecutive data flow and passes them to the filtering component. Then, an extraction and classification regarding the content of the frames takes place according to the [*source/destination*] value. Subsequently, an XML message conversion follows in order to achieve the appropriate compatibility with the corresponding grid applications and requests. The XML converter imports the sensor data to XML files according to predefined formats derived from the available XML database.

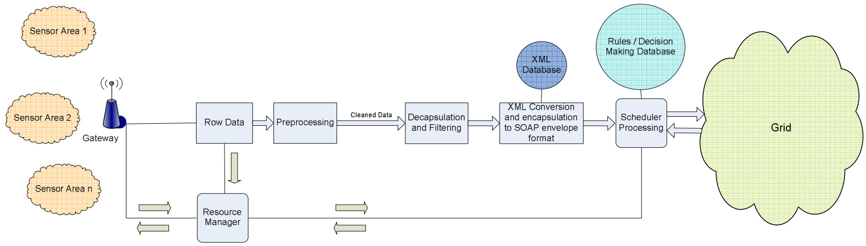

Fig. 2. Data flow in the proposed Sensor Grid framework

The XML output file depends on both the service/user class and the [*source/destination*] value. Hence, the final filtration of the overall message is accomplished at the scheduler module where the decision for the latter file format depends on the information by the QoS database where all rules for the grid services and the users take place. Along with the QoS database, the scheduler is directly communicating with the resource manager module which is also responsible for service class classification. Furthermore, the resource manager sends statistics for the average power utilization that has been observed in each sensor area and WSN to the scheduler in fixed time intervals. This information is derived by the MAC protocol that is used in each sensor area where an appropriate traffic monitoring of each sensor device (or sensor id) is achieved. Based on the aggregated resources that have been consumed in each sensor area, a classification of the available QoS is done and access is given or denied to grid applications according to their specific requirements.

The preprocessing and filtering components of the above mentioned framework implement PHY layer functionalities, as mentioned earlier. Therefore, the modeling of these components represents a task which is out of the scope of this paper. Our main interest has been placed on the modeling of an appropriate scheduler and a resource manager in order to provide energy awareness to sensor grids deployments as figure 3 shows.

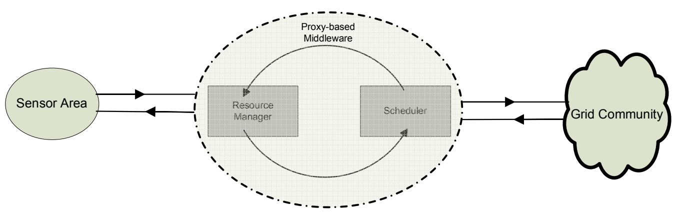

Fig. 3. Interaction of the main middleware components

3.2 Sensor Area

The sensor area represents the target interest of grid users in a sensor grid infrastructure. It contains a certain number (1 to N) of different WSNs; each WSN implements its MAC processes independently in order to serve grid calls. All WSNs in a sensor area communicate with the same gateway. The gateway is responsible for the coordination of communication within the WSN by sending signaling messages to all the nodes that participate in each grid call. In particular, upon a grid call request, the gateway notifies the requested node to measure the requested data information and all the intermediate nodes to the path from that node to the gateway. This latter procedure is accomplished via signaling messages which hold the included sensor ids after the instructed decision of the resource manager (as discussed later on). You may notice that each WSN in a sensor area could implement different MAC protocols in a fully distributed manner as long as the communication is accomplished via the same gateway. The proposed proxy-based middleware defines the routing strategy that should be followed, through an appropriate resource manager from a network perspective.

For the modeling of the WSN we consider a directed acyclic graph DAG (G, E) where all vertices correspond to the nodes of the network. The terms vertex and node will be used interchangeably in the sequel. An edge E between two vertices i and j, exist iff (if and only if) node i is adjacent to node j and can communicate directly with it. Thus, there is an edge $E_{i,j}$ iff $d_{i,j} \leq d_{eff}$, where $d_{i,j}$ denotes the distance between nodes i and j and d_{eff} denotes the effective maximum distance due to the transmission range of the wireless sensors. We also assume, without loss of generality, that the transmission range is the same for all the sensor nodes. The direction of the data flow is always from the polled node (transmitter) to the gateway (receiver) of the considered graph. Prior to the message exchanging procedure, the signaling mode takes place. The signaling mode is modeled by a graph DAG′ (G′, E′) with the opposite direction (from the gateway to the requested node and all intermediate nodes that will participate in the communication process).

Note that the above mentioned routing decision is accomplished in the network layer perspective. To provide multi-hop relay services in a WSN, or a sensor area in general, the resource management at the link layer and the routing at the network layer interact with each other. As the first step in our research, we consider separate designs at the resource management on routing and the resource management on MAC, and assume that a MAC protocol is already in place. How to achieve an optimal or suboptimal joint design of routing and resource management is very important issue for further research.

3.3 Scheduler

The scheduler module provides the interface of the middleware with the external grid community. Upon a grid call, the scheduler verifies the request id according to specific validation criteria which are stored in a user profile database. The authorization and the authentication of incoming grid calls are accomplished by consecutive interactions of the scheduler with the database. Each user has to register to the system in order to get access to the sensor areas. According to registered profiles the scheduler decides for the service class that could be supported from the system hereafter. If the requested service class matches the credentials of the associated user profile the procedure continues, otherwise there is a drop call event due to lack of necessary credentials.

More specifically, the system supports a fixed number of user groups, each with different access rights. User group entries can be defined as:

$$V_g = [\text{res\_util}, \text{ group\_thr}], \qquad g = 1,2,3,...,N \tag{1}$$

where V_g is a vector which contains the credentials for the specific user group g, res_util denotes the maximum resource utilization percentage per request per node for the current user group and group_thr denotes the maximum resource utilization per request per node when the available resources of a considered node are equal to the minimum allowed energy threshold of sensor node upon the current request.

Every user has a registered entry in one of the above mentioned user group profiles which is stored in the database in the following format:

$$V_u = [user\_id, V_g], \qquad u = 0,1,2,...,number\ of\ users \tag{2}$$

where V_u represents a vector which holds all the user registrations, user_id denotes an identifier unique for each user and V_g is a pointer to (1) which shows the service class that could be supported.

The service class is a factor corresponding to the number of services that the requested sensor node can serve upon a grid call arrival. As mentioned in the previous section, a sensor device can support multiple services at the same time, each associated with a transceiver, e.g. light, sound and humidity monitoring, temperature and vibration sensing. Each class denotes the number of services that can be served at the same time from a requested node. Without loss of generality, we assume for the rest of the paper that sensor node measurements for all kinds of supported services require the same power consumption level. A typical example of QoS classification and average resource consuming estimation is presented in Table 2, where five of the most popular sensor activities have been taken into consideration.

Table 2. Service class characteristics

QoS Classification	Types of provided services by sensor devices	Recourse Utilization (%) of sensor device per measurement
Class 0	Light Monitoring, Sound Monitoring, Humidity Monitoring, Temperature Sensing and Vibration Sensing	100
Class 1	Light Monitoring, Sound Monitoring, Humidity Monitoring, Temperature Sensing	80
Class 2	Light Monitoring, Sound Monitoring, Temperature Sensing	60
Class 3	Light Monitoring, Temperature Sensing	40
Class 4	Temperature Sensing	20

Other sensor activities, such as pollution measurements, could also be adapted to the proposed model. However, for simplicity reasons, the above mentioned five well-known sensor activities have been considered for the proxy-based middleware framework. The percentage of the average resource utilization accounts for specific energy thresholds that can be observed in a sensor node. Based on the requested service class thresholds and the information of the remaining power resources of each node in the given sensor area, the scheduler (communicating with the resource manager) classifies the availability of services and accepts or denies the grid user requests according to the network status.

Note that in this paper we have focused on the energy efficiency for middleware sensor grid deployments, hence all scheduler operations such as user authentication and authorization, grid call acceptance or drop call events, service and user group classification have been implemented from a power awareness point of view. Hence, the proposed scheme is termed as *power efficient*.

3.4 Resource Manager

In order to serve a valid grid request, a second-level control mechanism checks the current network status and the availability of the requested sensor nodes. The main role of the resource manager is the energy conservation of the entire monitored sensor network. The routing decision from the requested node/nodes to the gateway is taken according to the remaining energy of the sensor nodes and the energy consumption for the specific service class demands. Therefore, the resource manager finds optimal [*source/destination*] paths while extending the sensors' lifetime and preventing network partitioning. In particular, it keeps a record file containing all the monitored sensor ids with their respective remaining energy resources. It is also aware of all the adjacency links among the nodes within its sensor area.

For modeling purposes we consider a rectangular grid area where all the nodes are placed uniformly in the plane. Hence, the grid area is a [n ^{x} n] matrix, where n denotes the number of nodes in the sensor area. Let AdjL= [$n^{2\,x}$ n^2] matrix corresponding to all available links among neighboring nodes. We define the following indicator function:

$$AdjL_{ij} = \begin{cases} 1, & \text{if } i \text{ is adjacent to } j \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

where $AdjL_{ij}$ denotes the existence of an adjacent link from node i to node j. We also define:

$$P_c^j = N_S^c \ \hat{p}^j \tag{4}$$

where $\hat{p}^j$ denotes the power consumption for a single service measurement for node j, N_S^c denotes the maximum number of services of class c and P_c^j denotes the overall energy consumption for node j for service class c. Hence, each routing path can be expressed as:

$$Path_\xi = \sum_{j=1}^{m} [P_c^j], \qquad \xi \in \mathbf{Z}^+ \tag{5}$$

with respect to the power consumption within sensor area, where ξ stands for the number of available paths in a sensor area and m denotes the number of hops from the gateway to node j or vice versa. In order to optimize the overall energy conservation, the resource manager always selects the shortest path/paths since it maintains low-level energy thresholds in its monitored sensor area. Thus, it selects all the paths with minimum hop count m, as:

$$Sh\_P_x[] = \sum_{X \subseteq \Omega} \min_m (Path_\xi) \tag{6}$$

where Sh_P_x stands for the shortest path selection with respect to X which denotes a subset of all available shortest paths within a subset of Ω, where Ω denotes all the available paths from the gateway to node j or vice versa.

Due to the assumption of uniform distribution of the sensor nodes the network topology may provide several shortest paths with equal hop count m, for a given node j to gateway. A typical example is shown in figure 4 where node A can communicate with node B via one of its adjacent nodes denoted with dashed line. The latter diversity in path selection is provided due to symmetry nature of the [4 ^{x} 4] grid topology.

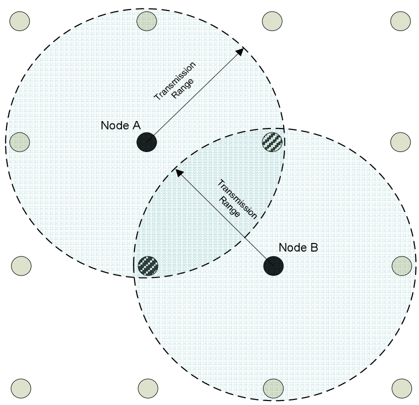

Fig. 4. Grid Topology Simple-Case Scenario

The resource manager (in collaboration with the scheduler), finds the optimal shortest path by solving the following linear program:

$$find \quad \min_{x \in X} Sh\_P_x \tag{7}$$

subject to

$$[Cur\_Sensor\_thr]_j - [Sensor\_thr]_j - P_c^j \geq 0, \quad \forall \ j \ in \ the \ path \tag{8}$$

$$[Res\_util]_u^{V_u} - [req\_C]_u \geq 0 \tag{9}$$

where Cur_Sensor_thr denotes the available power resources of node j and Sensor_thr denotes a minimum power level that a considered node must possess in order to participate in communication procedure. Finally, req_C denotes the requested service class upon a grid call arrival from user u.

Upon the selection of the optimal path, the resource manager sends to the gateway a sensor-id list which contains the requested sensor node and all the other nodes that participate in the selected path. In order to take into consideration the latter signaling cost, (8) is transformed to:

$$[Cur\_Sensor\_thr]_j - [Sensor\_thr]_j - P_c^j - sig\_P_c^j \geq 0, \quad \forall\ j\ in\ the\ path \quad (10)$$

where sig_ P_c^j denotes the signaling cost percentage with respect to P_c^j, for a packet transmission from the gateway to node j.

If there is no available shortest path fulfilling the criteria denoted by the scheduler and the resource manager, the above mentioned linear program is re-executed substituting C_n to C_{n+1} until all the available service classes are covered, in order to minimize the drop call probability.

4 Performance Evaluation

We have implemented the proposed framework in a JDK 6.0 environment. For our simulations we consider that the grid call arrival rate from each user group is defined by a Poisson process, according to the number of requests per minute of each user group. Analytical simulation user parameters are listed in Table 3. Each call is associated with a specific node-id from the sensor grid. For the selection from the grid, sensor nodes are statistically independent, identically distributed with unit variance. We therefore used a uniform distribution for the association of requested nodes for each call due to their equal selection probability. The sensor area is considered to be a WSN consisting of 100 equally positioned nodes (10^x10 grid plane) and the gateway is placed in the middle of the sensor area in order to maximize the path selection diversity and to avoid rapid network energy saturation.

Table 3. User group statistics

	Administrator	Government	Academic	Commercial	Unsubscribed
Number of Users	100	200	2000	4000	8000
Call Rate per user (*per minute*)	0.15	0.1	0.05	0.03	0.015
res_util (%)	100	100	80	60	20
group_thr (%)	0	5	10	15	20

We fix parameter $\hat{p}^j$ and sig_ P_c^j to be 0.23^x10^{-3} % and 0.023^x10^{-3} % respectively, according to the methodology followed in [10-12]. For the configuration of the proposed resource manager component (as described in subsection 3.4), we have implemented a breadth-first search in order to find the optimal shortest path/paths. A crucial benefit for the above mentioned decision is its direct response mainly due to the low-level complexity of the algorithm defined as O(|V|+|E|). Figure 5, shows the flowchart of the proposed scheme and its basic characteristics.

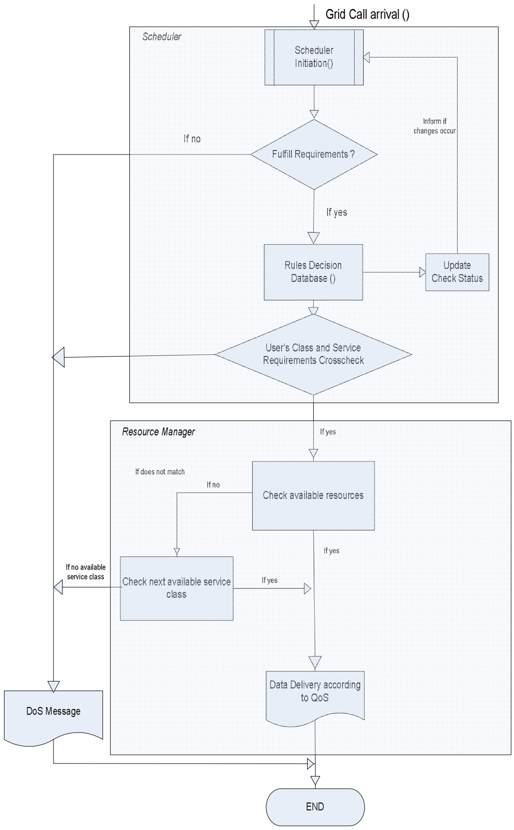

Fig. 5. Flowchart of the proxy-based Middleware

4.1 Simulation Results

In order to evaluate the effectiveness of the proposed scheme, a cross-reference scenario was necessary. We therefore compare our scheme to a secondary algorithm (it is termed *Res_Algorithm*) which employs only the resource manager component. Since scheduler is not adopted, it can not filter grid calls according to the "type of user" criterion. In other words, it does not provide any QoS classification. All user groups can get access to all available service classes. In addition, the resource manager of the secondary algorithm is not optimized, in the sense that it does not implement the linear program in (7), (9) and (10) for the shortest path selection. Instead, it selects randomly one of the shortest paths derived by (6). A tertiary algorithm (called *Sched_Algorithm*) employs the proposed scheduler component and the modified simple resource manager component as illustrated in the Res_Algorithm.

The main goal of the implementation of the two alternative schemes is to evaluate independently the importance of the proposed scheduler and the resource manager components.

Figure 6 shows the aggregated percentage of the energy consumption of the entire WSN as a function of the simulation time. The termination of the energy consumption lines means that there is no more available energy to serve requests. As expected the *Res_Algorithm* results in the shortest network lifetime because the absence of the Scheduler leads to much higher energy consumption for serving the requests of the less-privileged user groups. We can also observe that the energy consumption exhibits extremely sharp falls due to the fact that the Resource Manager does not perform the linear programming of functions 7-9. The result is that the energy is consumed linearly until it is enough only for low service classes and then until its complete exhaustion. Although the *Sch_Algorithm* controls better the energy consumed by the less privileged user groups, it still suffers from the sharp falls that shorten the network lifetime. The proposed architecture leads to the longest serving duration due to the more sophisticated implementation of the Resource Manager. When the available energy falls below 30%, the Resource Manager accepts requests from increasingly fewer user groups to reserve power for the more privileged users. Consequently the energy consumption degrades more gracefully and the lifetime of the sensor network is prolonged in favor of the more "important" users.

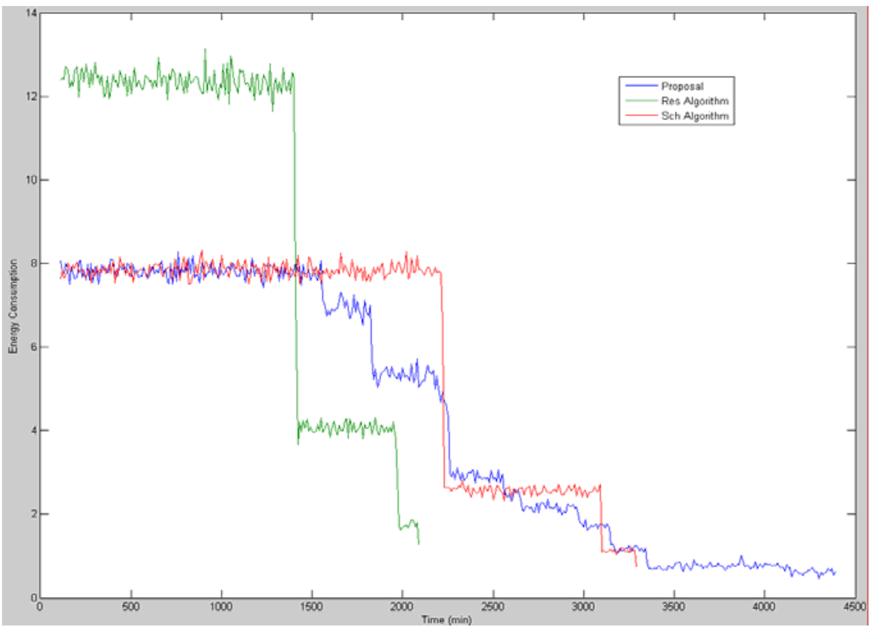

Fig. 6. The aggregated energy consumption percentage of the three simulated algorithms as function of the time

Figure 7 shows in more detail how the Proposed Algorithm affects the energy consumption and the accepted request rate per user group in comparison to the *Res_Algorithm*. When the *Res_Algorithm* is used, the percentage of the accepted requests and the energy consumed by a user group are connected through the equation $\frac{E_i}{\sum E_i} \simeq \frac{R_a^i}{\sum R_a^i} \simeq \frac{R_t^i}{\sum R_t^i}$, where E_i is the energy consumption, R_a^i is the rate of the accepted requests from group i and R_t^i is the overall request rate from group i. Namely the amount resources that a user group utilizes depends to its request rate and not to its role. As figure 7 indicates, the Proposed Algorithm achieves the desired QoS differentiation by accepting fewer requests from the Commercial and Unsubscribed user groups, giving priority to the more privileged users. As a result, the acceptance rates of the Administrator and the Governmental users increase by 132% and 46.5% respectively. The energy consumption change ratio is even larger because of the consumption restrictions that the Scheduler imposes to the lower user groups.

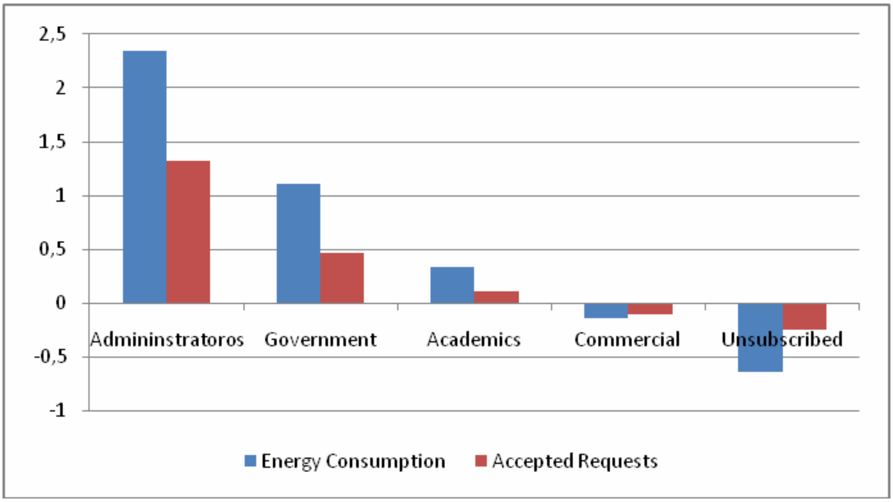

Fig. 7. The change ratio of the energy consumption percentage and the accepted request rate between the Proposed Algorithm and the Res_Algorithm

Although the simulation results illustrate the benefits from using the Resource Manager in conjuction with the Scheduler, the above scenario is only indicative. The logic of the Resource Manager and the Scheduler can be easily expanded to meet the requirements of different grid applications. As an example of such optimizations we can simulate a second test case that is similar to the previous but now we assume that the grid applications are not extremely time-sensitive. In this case the Scheduler can use a delay queue that caches the incoming requests for a short time interval before forwarding them to the Resource Manager. If there are multiple requests for the same node before the delay time timeouts, these requests can be translated to only one sensor-level request. The delay time can be a multiple of the interarrival request time and if a request is time-sensitive, a flag can be set to indicate that it should be served directly. Figure 8 depicts the perfomance of this delay queue with respect to the total number of accepted requests when using the Proposed Algorithm. We should note that for 19.5% consumed energy the network cannot serve more requests. The reason is that the nodes that communicate with the sink have consumed all their energy while the nodes' energy increases as we move further from the sink.

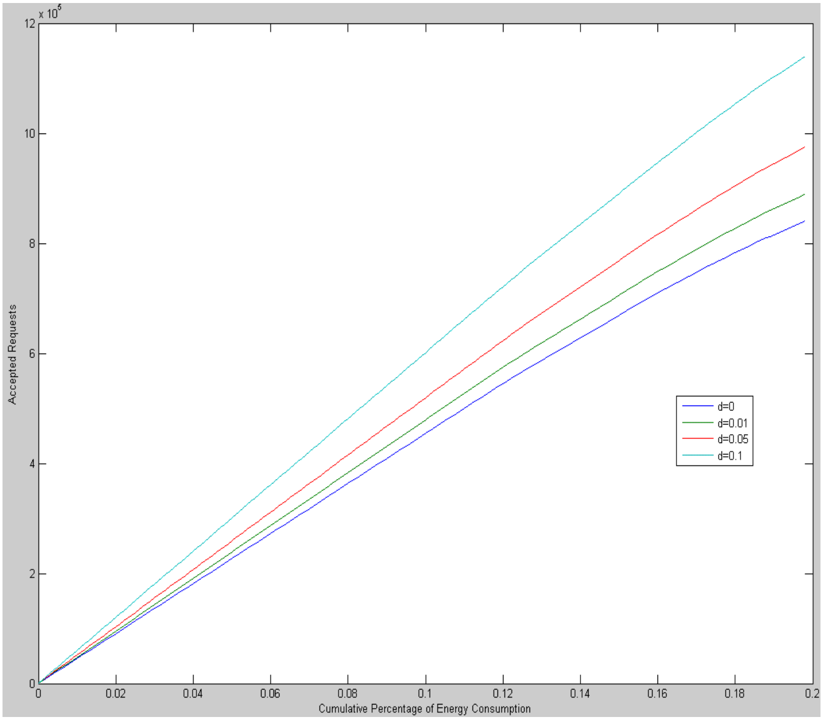

Fig. 8. The total number of accepted requests as a function of the cumulative percentage of consumed energy of the whole WSN

As expected, longer delays entail a larger number of served requests for the same level of energy consumption. However the choice of the appropriate delay depends on how timely an application should be. It should be also noted that the performance of the queue depends on two important parameters, the number of the nodes and the probability distribution for selecting a specific node. When the nodes are selected uniformly the delay queue performs worse for larger networks. On the contrary the efficiency of the delay queue increases if some nodes have a higher probability to be selected than others (e.g. following the Binomial distribution). This is usual if some phaenomena take place only in specific areas of the WSN. Finally, it is worth mentioning that the delay queue allows the less privileged user groups to obtain measurements even if the available energy is below their respective energy threshold. This can happen whenever two different requests for the same node are generated by two different user groups, a privileged and a less privileged one. The Scheduler will be responsible to extract the data from the reply to the privileged user group, that are allowed to be accessed by the less privileged.

4.2 Discussion

The major advantages of our work are as follows. Firstly, it is a scheme which combines efficient routing in WSN infrastructures and QoS classification in a personalization logic basis from an energy awareness perspective. Additionally, it is the first scheme which combines the above mentioned characteristics in an integrated platform designed specifically for sensor grid applications. Secondly, it is fully compatible with any sensor network deployment, in the sense that is placed in the middleware without any interaction with the MAC implementation of each WSN. It is also a distributed approach as each sensor area is managed separately by its associate proxy-based middleware. Thirdly, the low-level complexity of our scheme provides an essential benefit which is more than a prerequisite for a QoS-provisioned grid infrastructure, consisted of a dense grid community.

5 Conclusion

WSNs and Grid Computing are two promising technologies and both have been adopted into industry recently. Sensor grid deployments enhance a great potential of these technologies into a merged framework and due to that the research community has focused on innovative strategies to the field. A middleware architecture platform is a prerequisite for sensor grids in order to efficiently come through aggregated grid services and rapid user demands. In this paper, we discussed the most challenging issues for a proxy-based middleware scheme in order to cope with sensor grids and we also proposed a model which accepts grid calls according to specific service classes giving appropriate QoS on a personalization logic basis. Furthermore, the whole service handling and management framework is considered to be power-aware according to sensor network status.

References

1. Tham, C.-K., Buyya, R.: SensorGrid: Integrating Sensor Networks and Grid Computing. Special Issue on Grid Computing (2005), `http://www.gridbus.org/reports/sensor-grid.pdf`
2. Pantazis, N., Vergados, D.D., Miridakis, N.I., Vergados, D.J.: Power control schemes in wireless sensor networks for homecare e-health applications. In: Proceedings of the 1st international conference on PErvasive Technologies Related to Assistive Environments, Athens, Greece (2008)
3. Lim, H.-B., Teo, Y.-M., Mukherjee, P., Lam, V.-T., Wong, W.-F., See, S.: Sensor Grid: Integration of Wireless Sensor Networks and the Grid. In: The IEEE Conference on Local Computer Networks, pp. 91–99 (2005)
4. Franke, H.A., Koch, F.L., Rolim, C.O., Westphall, C.B., Balen, D.O.: Grid-M: Middleware to Integrate Mobile Devices, Sensors and Grid Computing. In: Third International Conference on Wireless and Mobile Communications (2007)
5. Rao, I., Imran, N., Khan, S., Huh, E.-N., Chung, T.: Adaptive and Reconfigurable ResOurce Management for Wireless Sensors using Grid Technology. In: 2nd International Conference on Communication Systems Software and Middleware, COMSWARE, pp. 1–5 (2007)
6. Gaynor, M., Moulton, S.L., Welsh, M., Lacombe, E., Rowan, A., Wynne, J.: Integrating Wireless Sensor Networks with the Grid. IEEE Internet Computing 8(4), 32–39 (2004)

7. Stathis, K., Kafetzoglou, S., Papavassiliou, S., Bromuri, S.: Sensor Network Grids: Agent Environments Combined with QoS in Wireless Sensor Networks. In: Third International Conference on Autonomic and Autonomous Systems (2007)
8. YuJie, Y., Shu, W.: The key research on integrating wireless sensor network with grid. In: Proceedings, International Conference on Wireless Communications, Networking and Mobile Computing, vol. 2, pp. 1477–1480 (2005)
9. Miridakis, N.I., Vergados, D.D., Anagnostopoulos, I., Douligeris, C.: A discussion on proxy-based middleware for Sensor Grid Architectures. In: 4th EGEE User Forum/OGF25 & OGF-Europe's 2nd International Event, Catania, Sicily, Italy (2009)
10. Crossbow, MICA2 wireless measurement system. Datasheet available from, `http://www.xbow.com/products/Product_pdf_files/Wireless_pdf/MICA2_Datasheet.pdf`
11. Piotrowski, K., Langendoerfer, P., Petter, S.: How public key cryptography influences wireless sensor node lifetime. In: The proceedings of 4th ACM Workshop on Security of ad hoc and Sensor Networks, pp. 169–176 (2006)
12. Bouabdallah, F., Bouabdallah, N., Boutaba, R.: On Balancing Energy Consumption in Wireless Sensor Networks. IEEE Transactions on Vehicular Technology 58(6) (2009)
13. Gaynor, M., Moulton, S.L., Welsh, M., LaCombe, E., Rowan, A., Wynne, J.: Integrating Wireless Sensor Networks with the Grid. IEEE Internet Computing 8(4), 32–39 (2004)
14. Tuecke, S., Czajkowski, K., Foster, I., Frey, J., Graham, S., Kesselman, C., Maquire, T., Sandholm, T., Snelling, D., Vanderbilt, P.: Open Grid Services Infrastructure (OGSI) Version 1.0, Global Grid Forum (2003)