On the Design of a Self-Managed Wireless Sensor Network

Linnyer B. Ruiz, Thais R. M. Braga, Fabrício A. Silva, Helen P. Assunção, José Marcos S. Nogueira, and Antonio A. F. Loureiro, Federal University of Minas Gerais

ABSTRACT

During the last decade, there was a great technological advance in the development of smart sensors, low-power processors, and wireless communication protocols that when put together create a wireless sensor network (WSN). Smart, autonomous, and self-aware: that is the ultimate vision for WSNs. The success of this vision depends fundamentally on the self-management solutions. This work deals with this challenge: to provide a self-management solution for a WSN that monitors temperature and evaluates fire risks. We focus on self-organization, self-configuration, self-service, and self-maintenance. In particular, we propose service negotiation policies to demonstrate the self-service concept. The results reveal that the management solution can promote the productivity of network resources and the quality of the provided services.

INTRODUCTION

Wireless sensor networks (WSNs) are an emerging technology that promises unprecedented ability to monitor, instrument, and eventually control the physical world. WSNs, in general, consist of a large number of inexpensive wireless devices (sensor nodes) densely distributed over the region of interest. Sensor nodes have wireless connectivity and can be tied to a backbone network such as the Internet. They are typically battery-powered, and equipped with a processor, memory, and a variety of sensing modalities (e.g., acoustic, meteorological, seismic, and infrared). A sensor node tends to be designed with small dimensions, and this size limitation imposes restrictions on its resources (e.g., energy, communication, and processor capacities). The logical component of a sensor node is the software that runs in its computational unit. The interconnection of sensors through wireless communication networks, with the goal of performing a larger sensing task, will revolutionize the way information is collected and processed [1].

WSNs can be used in different applications for monitoring, tracking, coordinating, and information processing. For instance, sensor nodes can be dropped over remote areas (oceans, volcanoes, rivers, forests, etc.) and self-organize themselves to form an ad hoc wireless network that collects data of interest, performs local processing, and disseminates information to one or more access points (APs). They are expected to operate autonomously for periods of time ranging from days to years. In many applications, sensor nodes may not be easily accessible because of the locations where they are deployed or the large scale of such networks. In both cases, network maintenance (by technicians) for reconfiguration, energy replenishment, or technical problems becomes impractical. Therefore, a WSN must be able to operate under very dynamic conditions.

The success of this vision depends fundamentally on self-management solutions. Thus, a WSN must perform, besides services related to the application, management services with the goal of promoting network productivity and quality of service (QoS). However, the task of building and deploying self-management solutions in environments where there will be tens of thousands of network elements with particular features and organization is not trivial. This task becomes harder due to the resource restrictions of these unattended sensor nodes.

In [2] we proposed an architecture for WSN management called MANNA, and in [3] we discussed how this architecture is based on the self-management paradigm (i.e., systems manage themselves without direct human interference). The goal of this work is to design and analyze a self-management solution to a hierarchical heterogeneous WSN that performs fire risk monitoring, using some automatic services (self-organization, self-configuration, self-service, self-knowledge, self-awareness, and self-maintenance) and a decentralized management approach. The main contribution of this work is the development of a self-managed WSN that can define self-organization and self-maintenance polices using a well defined language called PONDER [4]. Furthermore, this self-management solution uses a Voronoi diagram to identify redundant nodes, calculates fire risk using the Angstrom index and attributes priorities to its messages according to this index, per-

0163-6804/05/$20.00

The MANNA architecture was proposed to provide integrated management solutions for different WSN applications. It provides a separation between application and management functionality sets, making integration of organizational, administrative, and maintenance activities possible for WSNs.

forms service negotiation through the use of a state machine, and reconfigures the hardware of its components through dynamic power management (DPM) functions. Simulation results reveal that the self-management solution proposed can save energy while fulfilling QoS requirements of the application, such as fast message delivery in case of an event (fire) occurrence.

The remainder of this work is organized as follows. We define the application chosen as a case study. We present an overview of the MANNA architecture. We describe how the self-management solution has been designed considering the application features. We discuss the simulation approach and the results, respectively. We then present our concluding remarks.

Application Scenario

In this work we consider as our case study the problem of temperature monitoring and fire risk evaluation in a remote area. This application represents a class of WSN applications with enormous potential benefits to the whole scientific community and society. In Brazil there are 34 national parks, 25 ecological reserves, 20 biological reserves, and 38 national forests, totaling approximately 28 million hectares. Instrumenting natural spaces with networked sensor nodes can prevent fire and advise related staff (e.g., firefighters) about risks.

The Meteorological Studies and Research Division, which is part of the Meteorology Service of Brazil's Agriculture Ministry, proposed two equations to estimate the degree of fire danger that are considered more viable for Brazil's climatic and structural conditions: the Angstrom and Nesterov indices. There is also another index developed in Brazil, the Monte Alegre formula, which is based only on humidity values. The Nesterov index has as input variables the temperature and air saturation deficit. The latter is calculated based on the maximum water vapor pressure and real water vapor pressure. On the other hand, the Angstrom index provides a more precise result than does the Monte Alegre formula, and also collects less data and performs less computation than does the Nesterov index.

The Angstrom index (I) involves only the percentage of relative humidity (H) and the air temperature in degrees Celsius (T). Sensor nodes collect temperature data and compute the fire risk using a relative humidity value equal to 60 percent, which corresponds to relative humidity of a *cerrado* region (a kind of Brazilian biome where fire incidences frequently occur). The Angstrom index is defined as follows: $I = 0.05\text{H} - 0.1(T - 27)$. The Angstrom index is used to determine the areas and moments with a greater risk of fire occurrence. Fire risk is indicated through the following scale: when $I > 2.5$ fire conditions are unfavorable. If $2.0 < I \leq 2.5$, fire conditions are favorable, and if $I \leq 2$, fire occurrence is very likely.

Considering the management solution proposed in this work, delivering data as quickly as possible whenever fire occurrence is very likely is one QoS requirement expected to be provided. Any energy saving strategy that might be in use by the network should be ignored in this case.

Description of the MANNA Architecture

WSNs differ from traditional computer networks in many aspects. In general, they have a large number of distributed elements that operate in remote areas without any human interference and have severe energy restrictions. Besides, they present dynamic topology even though nodes are stationary (nodes can be destroyed, or energy, security, calibration, and communication faults may occur). In a WSN, a network element failure is common, not an exception. WSNs and sensor node architectures are completely application-dependent. Thus, the management solution should try to be "compatible" with the type of application being managed. The WSN aspects presented, and many others that can be found in [5], impose design challenges and research opportunities.

The MANNA architecture was proposed to provide integrated management solutions for different WSN applications. It provides a separation between application and management functionality sets, making integration of organizational, administrative, and maintenance activities possible for WSNs. The approach used in the MANNA architecture introduces a novel organization for WSN management considering the two well-known management dimensions, functional area management and management levels, and proposing a novel dimension called WSN functionalities (Fig. 1). The traditional management dimensions were revisited from a WSN perspective. Considering that WSNs are application-dependent and observing the characteristics of various WSN applications, five main WSN functionalities were established: configuration, sensing, processing, communication, and maintenance. The intersection of the three planes (dimensions) defines a cell. Each cell contains a set of management functions. One or more management functions can fit into one or more cells of the cube. Management services are executed through a set of these functions, and the conditions to execute them can be established through the use of policies. The use of these management dimensions is a good strategy to deal with complex management situations.

From its three-dimensional organization, the MANNA architecture defines a list of management functions and services that can be performed automatically in order to design self-managed WSNs [3]. In this case, the management services can be executed with little or no human interference.

Managing the Network Using MANNA Architecture

In the scope of this work, we have developed a management solution by implementing some self-management services defined by the MANNA architecture such as self-knowledge, self-awareness, self-service, self-organization, self-configuration, and self-maintenance. The self-knowledge service allows an entity to know itself (i.e., its internal components, current state,

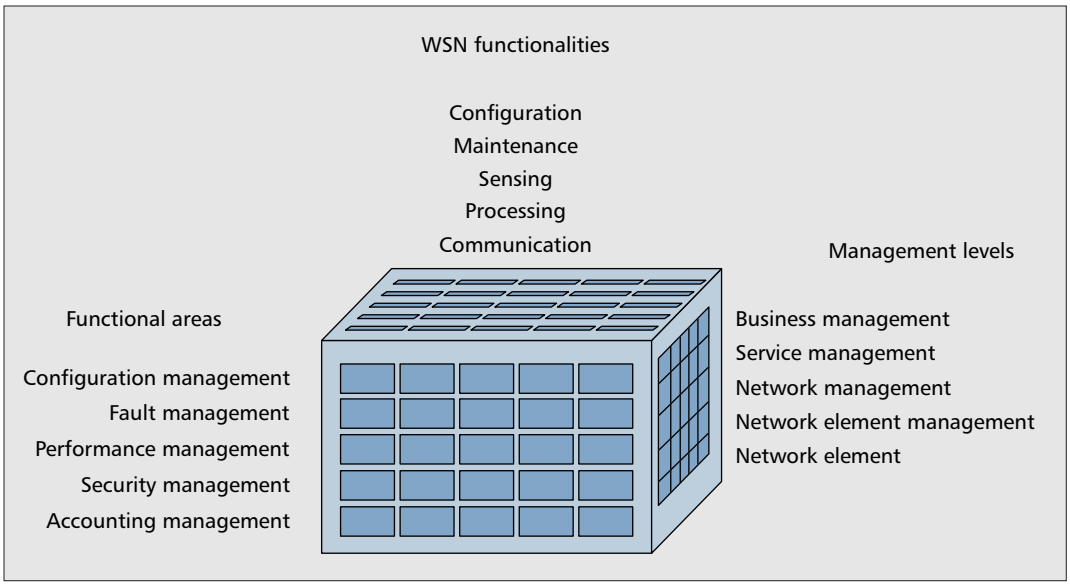

Figure 1. *Three-dimensional organization for WSN management.*

DPM is an effective tool to reduce the power consumption of a system without significantly degrading its performance. The management application is divided into three phases: planning, installation, and operation.

maximum capacity, etc.). When an entity implements the self-awareness service, it is able to know its environment and the context that surrounds its activities, and act accordingly. Note that the implementation of these two services is essential in the design of a self-management solution. Self-service allows a WSN entity to negotiate sensing, processing, and communication services. The network produces and delivers its own data. Regarding self-service, we have included service negotiation in our management approach. The basic idea behind service negotiation is to change the behavior of nodes according to environmental conditions or the behavior of the observed phenomenon. We have designed a self-managed WSN, also including DPM functions [6] whereby a sensor node can either shut down (entirely or some of its components), if it is considered a redundant node, or have its internal state changed according to the application behavior. DPM is an effective tool to reduce the power consumption of a system without significantly degrading its performance. The management application is divided into three phases: planning, installation, and operation, which are explained in the following.

PLANNING

Planning is executed through different functions that are performed before network bootup in order to decide the node architecture, number of nodes, distribution type, deployment type, organization type, protocols, and so on. Based on [7], we have decided to implement a network organized into clusters (i.e., a hierarchical WSN), which has nodes with different hardware capabilities (heterogeneous WSN) and with a high number of nodes per area unit (dense WSN).

Each network cluster must have a cluster head. In this work, the cluster heads have higher hardware capacities. Each node was programmed to perform application and management tasks. Small management agents were embedded into the common nodes, and the managers were embedded into cluster heads. Each manager was responsible for a subnetwork (cluster) and was able to negotiate with other managers. This is called manager-to-manager (M2M) architecture. The management entities exchange management information using a simple network management protocol called MannaNMP [5]. Traditional network management protocols, like Simple Network Management Protocol (SNMP), are not suitable for WSNs. Their centralized approach may lead to message implosion at nodes near the AP. Furthermore, they do not have scalability and features for efficient resource usage, which are key aspects in WSN design. Therefore, we have proposed MannaNMP, which considers the specific characteristics of WSNs.

Figure 2 presents an example of a hierarchical heterogeneous WSN, where agents and managers reside at common nodes and cluster heads, respectively. The AP is the point through which the network exchanges information with the outside world, that is, the way the observer can communicate with network nodes.

INSTALLATION

Once started, nodes perform a number of initialization routines (node setup), such as internal node self-test, health status determination, and sensor calibration. They also launch any procedures of configuration management at the network element level that have been preprogrammed to reflect specific application requirements and expectations [5]. For simplicity and convenience, it is assumed that each node knows its own location. Moreover, they are static and use omnidirectional antennas.

Main Services Implemented — During network bootup, the following services are performed: self-organizing, node scheduling management, and self-configuration.

Self-Organizing — This service is responsible for organizing nodes into groups in order to make the network scalable and robust. The problem of WSN self-organization could be

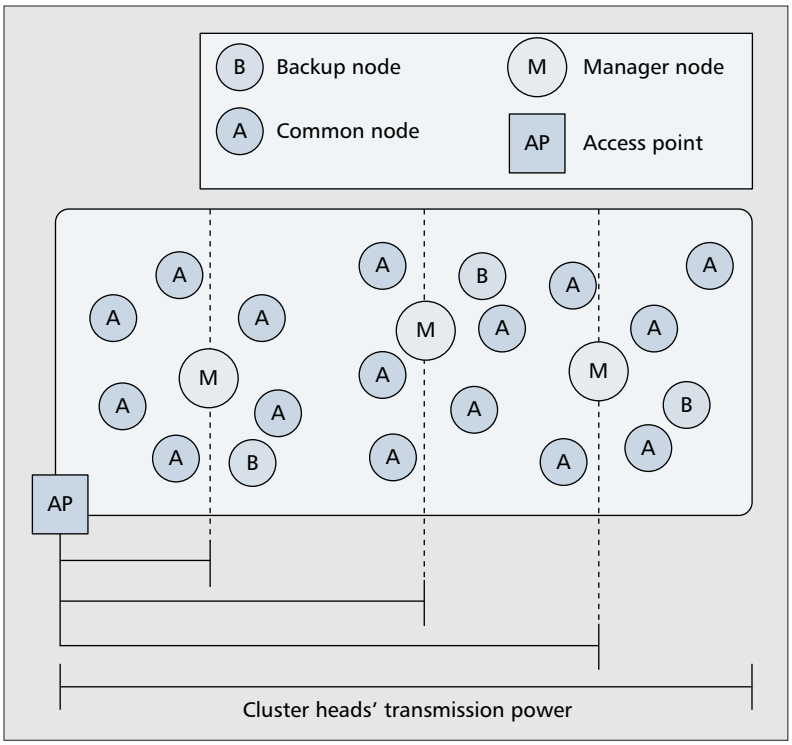

Figure 2. *Example scenario.*

solved using a global vision of the entire network. In hierarchical WSNs, a manager with this global vision could define which nodes would participate in each group, and after that indicate a cluster head for that group. Note that even using this centralized approach, the WSN would still be self-managed, since it makes its own decisions without human interference. In the experiments performed in [7, 8], autonomic and centralized management solutions were evaluated. This management approach extends the lifetime of the WSN, since the computation is performed outside the network, saving energy of sensor nodes. However, this solution may not be efficient for some applications if we consider some metrics such as packet loss and delay due to the large amount of traffic that may be generated in response to operation request and notification emissions.

In this work we have implemented and evaluated a decentralized management approach. This approach has the advantage of being scalable and minimizing the traffic generated by management. The goal is to develop a decentralized self-organization service based on policies, in which managers interact among themselves (M2M approach) in a cooperative fashion to achieve a desired overall goal: to form groups of nodes, control network density, and keep the coverage of the WSN area.

Controlling Density and Keeping the Coverage Area — Coverage area maintenance is a self-maintenance service that uses the density control function to identify nodes that can be administratively put out of service in order to reduce congestion, collision, and energy waste (i.e., redundant sensor nodes). In a dense network the management takes the redundant nodes out of service temporarily, balancing the network. In order to do so, the service uses DPM functions, shutting down the redundant nodes for a specific amount of time. Given that sensor nodes can fail, run out of energy, or be destroyed, sparse areas may appear (i.e., areas not completely monitored). When a sparse area is identified, the service tries to balance the network again, activating one or more backup nodes present in the region if available. In [8] we used the Voronoi diagram to identify backup nodes, considering a centralized management approach. In this work we implemented a noncentralized approach where managers installed in the cluster heads calculate the Voronoi diagram. Notice again that this is an autonomic implementation of the maintenance service, since only the network nodes, specifically the cluster heads, decide how and when to act, without any human interference.

Given a subnetwork comprised of N common nodes that are near a higher-capacity node, the Voronoi diagram algorithm builds, for each common node, a corresponding region called a *Voronoi area* such that any point inside a node's area is closer to this node than to any other node present in the subnetwork. A node that has a Voronoi area considered small by the management application, according to the sensing range, can be viewed as a redundant node and therefore taken out of service since certainly there are other nodes in its neighborhood with a similar sensing range. The sensing range will never be smaller than the Voronoi areas generated.

A manager cannot change the administrative state of a common node before negotiating with other managers.

Self-Configuration — Sensor nodes execute the self-configuration management service to change parameter values in order to adapt themselves dynamically to changing conditions or states of the network. In this work the cluster heads change their transmission power configuring their communication range according to their distances from the AP and common nodes of their clusters. The transmission power is set considering that the cluster head must reach the AP and all nodes in its cluster.

Defining Policies — The implementation of the services listed above is based on a set of policies that describe the desired behavior of each sensor node according to local information.

In this work we have chosen the PONDER language [4] to describe the proposed policies. PONDER is a declarative object-oriented language that can be used for the specification of both security and management policies. There are two types of policies defined by PONDER: authorization (A) and obligation (O). The former defines which activities a subject can perform on a set of target objects, and the latter shows which activities a manager or agent must or must not perform, also on a set of target objects. When it comes to huge systems, it is not feasible to specify policies to individual objects. In this case objects are grouped into domains to which policies are designed. To the application simulated in this work, we have identified two

id	*mode*	*[trigger]*	*subject*	*action*	*target*	*[constraint]*

P01 O+ **AT** NETWORK START TIMER H / /*Nodes became cluster heads*/0 H;
P02 O+ **AT** NODE START TIMER L / /*Broadcast of Localization Messages*/0 H;
P03 O+ **AT** INITIAL GROUP TIMER H / /*Form group, execute Voronoi and send Initial Group Messages to all cluster heads*/0 H;
P04 O+ **ON** bothRedundantsConflictDetection() n:H / /*Dispute node by number of redundants per group or by localization*/0 m:H;
P05 O+ **ON** bothActiveConflictDetection() H / /*Dispute node by number of redundants per group or by localization*/0 H;
P06 O+ **ON** activeRedundantConflictDetection() H / /*Cluster Head who identifies the node as redundant keeps it*/0 H;
P07 O+ **ON** recvMessage(Initial Group Message) n:H / /*Starts negotiation*/0 H **WHEN** n.alreadyNegotiated 121 FALSE;
P08 O+ **ON** decideCon_ict() n:H / /*Send Negotiation Message*/0 @/H **WHEN** n.hasNodesToRequest 121 TRUE;
P09 O+ **ON** recvMessage(Negotiation Message) H / /*Update Lists*/0 H;
P10 O+ **ON** DecideConflict() n:H / /*remove node from list*/0 H **WHEN** n.loseNodeContest 121 TRUE;
P11 O+ **ON** recvMessage(Initial Group Message) H / /*Negotiate according to messages order of arrival*/0 H;
P12 O+ **AT** FINAL GROUP FORMATION TIMER H / /*Send Paternity Message*/0 L/3 Cluster head id4 Group;
P13 O+ **AT** DENSITY CONTROL TIMER H / /*Send Density Message*/0 L/3 Cluster head id4 Group/RedundantNodes;

Table 1. *Policies definition. H:* HighCapacityNodes *and L:* LowCapacityNodes.

main domains. They are the *HighCapacityNodes* domain, representing the most powerful nodes of the network, and the *LowCapacityNodes* domain, representing the common nodes of the network. Table 1 presents the self-organization and self-configuration policies described through the PONDER language, obtained from [9].

Operation

We have implemented some management services (self-knowledge, self-awareness, self-service, self-maintenance) that are executed during the operation phase. Each message disseminated by the common nodes has a priority value (high or low, depending on the fire risk). When the cluster head receives a message, it evaluates its priority. When low-priority messages arrive, the cluster head computes the average of them and waits for a 120 s interval to send the result to the AP. If it receives a high-priority message, the average computed so far is discarded and the cluster head starts to aggregate high-priority messages during 5 s, when at last it sends a message containing all data to the AP.

The negotiation service proposed is characterized as the trade-off addressed by the sensor nodes, in which they need to adjust sensing, processing, and disseminating rates according to application and environment conditions. Thus, service negotiation is actually a dynamic adjustment of the network self-service. Summarizing, the common nodes are programmed to negotiate their services based on their states and the application behavior. Note that in order to perform service negotiation, it is very important for the nodes to know themselves and the environment that surrounds them. This negotiation is represented by a state machine (Fig. 3). The state machine transition policies are described below.

State-Transition Policies — States 0 to 3 are defined as follows:

• **State 0**: This means that a fire is occurring over the monitored area (fire index is below 2.0). Nodes in this state present a 10 s sensing interval, increasing the amount of information sent to their managers. When the fire is over and the temperature starts to decrease, the measurement of three consecutive fire indices higher than 2.0 makes nodes perform a state transition. If at least one of these three measurements is between 2.0 and 2.5, nodes must reach State 1. However, if all three measurements present indices higher than 2.5, nodes can go to State 2.

• **State 1**: This indicates that there is a fire risk in the monitored environment (fire index is higher than 2.0 and equal to or lower than 2.5). In order to perform a more accurate monitoring of the region, the sensing interval is set to 30 s. If the temperature keeps going higher, a transition to State 0 will be performed. If the temperature lowers, nodes will measure an index higher than 2.5, and after three consecutive measurements with index values like this, nodes will move to State 2.

• **State 2**: This state indicates that currently there is no fire risk in the monitored environment. It is the initial state, presenting a 45 sensing interval. Whenever a fire index equal to or smaller than 2.0 is computed, the nodes make a transition to State 0 and start to sense the environment with a 10 s sensing interval. If a fire index higher than 2.0 and equal to or lower than 2.5 is computed, the state transition is from

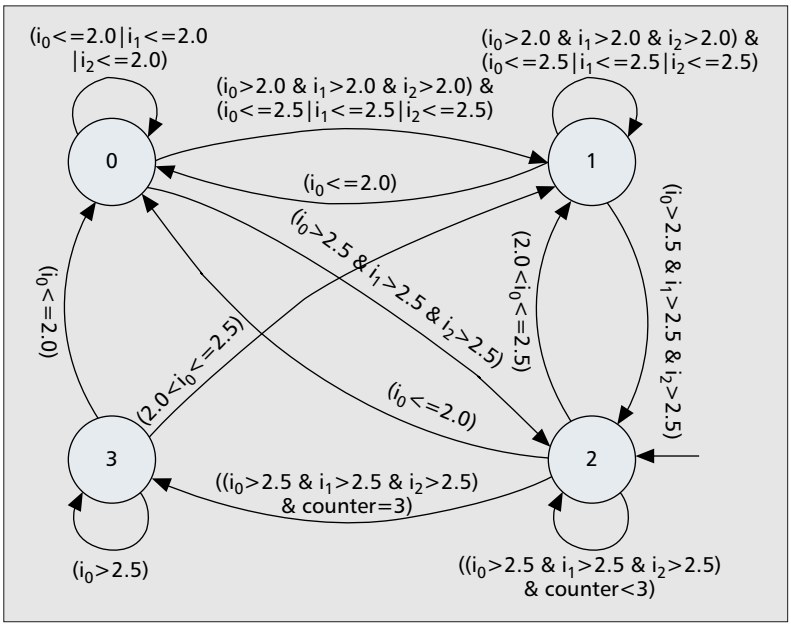

Figure 3. *The state machine describing the service negotiation.*

	Common nodes	Cluster heads	
		With power control	Without power control
Scenario 1	0.25 J	9.34 J	24.13 J
Scenario 2	0.32 J	10.19 J	30.20 J

Table 2. *Energy consumption in joules.*

State 2 to State 1, and the sensing interval is 30 s. If the nodes are in State 2, and three fire index measurements exceeding 2.5 are performed, they make a transition to State 3, where the sensing interval is 60 s.

• **State 3**: This state is reached only when there has been some time in which nothing happens over the monitored environment. The sensing interval here is 60 s. It can be observed that if a fire index below 2.0 is measured, nodes perform a state transition to State 0, and if the index is higher than 2.0 and equal to or lower than 2.5, the transition will be to State 1.

Simulation Approach

The self-management solution described earlier was evaluated through simulation using the Network Simulator 2 (NS-2) and a framework defined by a set of classes that extend NS-2 to simulate sensor networks. Our aim is to evaluate the efficiency and impact of the self-management solution over a WSN. To achieve this goal, two scenarios were defined and simulated.

Scenario 1: Planning and installation services are implemented. Besides, in this scenario the common nodes implement the service negotiation presented in Fig. 3.

Scenario 2: Planning and installation services are also implemented. However, the operation services are simplified, and service negotiation is not implemented. In this scenario the common nodes collect temperature data, compute the fire risk, and send it to the cluster head, which aggregates the received data. There is no change in nodes' behavior. By aggregation we mean the process of putting all data together, without any kind of processing. For instance, if there are N data of size S, the aggregated data will have size $N \times S$.

The simulation was performed using 12 cluster heads and 144 common nodes randomly deployed according to a uniform distribution, in a rectangular area of size 145 m × 112 m. The transport and medium access control (MAC) protocols used were UDP and IEEE 802.11, respectively. Each simulation ran for 2500 s and was repeated 33 times. The results presented next refer to the average of the obtained values. The characteristics of common nodes were configured according to Mica Motes [10] sensor nodes, with a transmission range of 70 m, a sensing range of 10 m, and a bandwidth of 28.8 kb/s. All of them collect data (sensing) periodically and disseminate it continuously. The characteristics of cluster heads, the most powerful nodes, were based on the WINS [11] sensor node, with a transmission range of 250 m and a bandwidth of 100 kb/s. They disseminate data periodically. The battery capacity of cluster heads and common nodes were 100 J and 10 J, respectively. The AP was deployed at coordinates X = 0 and Y = 0 of the area.

Two different types of processing were evaluated, each corresponding to a simulated scenario. In both scenarios the transmission power control (self-configuration) is implemented. In the network layer, a broadcast mechanism offered by the NS-2 simulation tool was used, in which all nodes perform only a single-hop communication. In this case a message reaches its destiny or is lost. The management protocol used in the application layer is MannaNMP because, as seen before, traditional protocols are not suitable for WSNs. The metrics used to evaluate the proposed solution are energy consumption, amount of collected/delivered data, and arrival time of the data at the AP.

Simulation Results

In this section we present the simulation results obtained and our evaluation of them.

Energy Consumption

In this section we have two goals. The first is to show that service negotiation is an interesting solution in terms of energy consumption when compared to aggregation processing. Table 2 presents the energy consumption for both scenarios. We can observe that the management solution of scenario 1 consumes less energy than scenario 2. Despite sending more data during a fire, for most of the simulation time scenario 1 sends smaller data (in bytes), since it processes it before dissemination and sends just the average of the received data.

The second goal is to show that self-configuration (transmission power control) is a good strategy to save energy. To evaluate this, we have simulated the same scenarios without this service. It is clear that this management service is very interesting, saving energy of cluster heads as shown in Table 2.

Sensed Data Flow

Table 3 presents the amount of data sent by common nodes and received by cluster heads at the AP. Each data unit (DU) represents a 8-byte structure that encapsulates sensed data. Observing Table 3, we notice that the amount of data sent by common nodes in Scenario 1 is smaller than in Scenario 2. The same is true for the amount of data received by the cluster heads and AP. This means that service negotiation reduces the amount of data in the network. The common nodes in Scenario 1 adjust their behavior according to the application behavior.

Data Arrived at the Access Point

An unexpected event was set to happen at t = 500 s of the simulation time. Actually, temperature variation has begun at different points of the monitored area. Figure 4 presents the time the sensed data arrived at the AP. We can observe in Fig. 4 that the self-managed WSN

implemented in Scenario 1 is more efficient in fire detection than in Scenario 2, which does not implement service negotiation. The self-managed WSN of Scenario 1 is able to delivery data with different priorities. In this way, the AP is informed about the fire earlier in Scenario 1 than in Scenario 2. We also notice that Scenario 1 presents a greater number of samples during the fire event. This allows the observer to have more accurate information.

Figure 4 is also important to show that even with sensor nodes sensing and disseminating data at higher rates (smaller intervals) in Scenario 1, generating a great amount of traffic at the network, the fire detection information arrives at the AP with a reasonable delay. The redundant data generated may cause congestion and overhead at the network. However, this redundancy is important to achieve the QoS requirements, since it allows the AP to be aware of the fire earlier and with more accuracy (more samples).

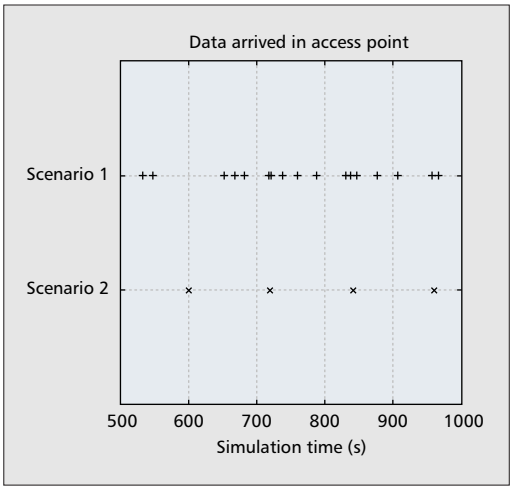

Figure 4. *Data arriving at the access point.*

	Common nodes	Cluster heads	Access point
Scenario 1	4584 DUs	4580 DUs	493 DUs
Scenario 2	5967 DUs	5964 DUs	5747 DUs

Table 3. *Sensed data flow measured in data units.*

Conclusions

The task of designing management solutions for specific problems in WSNs is not trivial. The integration of these solutions to promote the network productivity and QoS is even a greater challenge. This work has overcome this challenge by implementing a self-managed WSN using management services proposed by the MANNA architecture. Management of a wireless sensor network poses new research challenges, and we are able to propose novel solutions to some of these problems.

In this work we have evaluated the service negotiation of a self-management solution for a WSN designed to detect fire in a given area. In this case, sensor nodes change their sensing and disseminating intervals according to the risk of fire computed. In other words, sensor nodes collect and disseminate data at higher rates when there is a risk of fire. Otherwise, these rates are reduced. Two scenarios, one that implements service negotiation and one that does not, were simulated. The results reveal that the service negotiation solution, in addition to saving energy and increasing network lifetime, allows the access point to be notified about the fire earlier and with more accuracy.

Acknowledgment

The development and studies described in this article were completed as part of the Sensornet and Mannasim projects (http://www.sensornet.dcc.ufmg.br), funded by CNPq/Ministry of Science and Technology (Scientific and Technological Development Council). Some scholarships were given by CAPES/Ministry of Education (Coordination for the Improvement of Higher Education Personnel Foundation).

References

[1] D. Estrin, R. Govindan, and J. Heidemann, "Embedding the Internet," *Commun. ACM*, vol. 43, no. 5, May 2000, pp. 39–41.
[2] L. B. Ruiz, J. M. S. Nogueira, and A. A. F. Loureiro, "MANNA: A Management Architecture for Wireless Sensor Networks," *IEEE Commun. Mag.*, vol. 41, no. 2, Feb. 2003, pp. 116–25.
[3] L. B. Ruiz, J. M. S. Nogueira, and A. A. F. Loureiro, "Sensor Network Management," *Handbook of Sensor Networks: Compact Wireless and Wired Sensing Systems*, M. Ilyas and I. Mahgoub, Eds., Ch. 3. CRC Press, 2004.
[4] N. Damianou *et al.*, "The Ponder Policy Specification Language," *Proc. Int'l. Wksp. Policies for Distrib. Sys. and Net.*, vol. 1995/2001, Bristol, U.K., Ja.n 2001; Springer-Verlag, *LNCS*, pp. 18–38.
[5] L. B. Ruiz, "MANNA: A Management Architecture for Wireless Sensor Networks," Ph.D. dissertation, Federal Univ. of Minas Gerais, Belo Horizonte, MG, Brazil, Dec. 2003.
[6] A. Sinha and A. Chandrakasan, "Dynamic Power Management in Wireless Sensor Networks," *IEEE Des. and Test of Comp.*, vol. 18, no. 2, 2001, pp. 62–74.
[7] L. B. Ruiz *et al.*, "On Impact of Management in Wireless Sensors Networks," *Proc.9th IEEE/IFIP Net. Ops. and Mgmt. Symp.*, Seoul, Korea, Apr. 2004, pp. 657–70.
[8] M. A. M. Vieira *et al.*, "Scheduling Nodes in Wireless Sensor Network: A Voronoi Approach," *Proc. 28th Annual IEEE Local Comp. Net.*, Bonn/Konigswinter, Germany, Oct. 2003, pp. 423–29.
[9] F. A. Silva *et al.*, "Designing a Self-organizing Wireless Sensor Network," *Proc. 1st Int'l. Wksp. Mobility Aware Tech. and Apps.*, Florianópolis, SC, Brazil, Oct. 2004, Springer-Verlag *LNCS*, pp. 186–95.
[10] J. L. Hill and D. E. Culler, "MICA: A Wireless Platform for Deeply Embedded Networks," *IEEE Micro*, vol. 22, no. 6, 2002, pp. 12–24.
[11] G. J. Pottie and W. J. Kaiser, "Wireless Integrated Network Sensors," *Commun. ACM*, vol. 43, no. 5, 2000, pp. 51–58.

Biographies

Linnyer Beatrys Ruiz (linnyer@dcc.ufmg.br) is an associate professor of electrical engineering at the Federal University of Minas Gerais (UFMG), Brazil. Her areas of interest and research include computer networks, telecommunications and computer network management, and wireless sensor networks. She received a Ph.D. degree in computer science from UFMG, an M.Sc. degree in electrical engineering and industrial information from the Federal Center of Technological Education of Paraná (CEFETPR), Brazil, in 1996, and a B.Sc. degree in computer engineering from PUCPR. She held a post-doctoral position at UFMG, 2004. She is an expert in telecommunications management networks. Since 1993 she has participated in and coordinated research groups on TMN. Currently, she is coordinating the WSN

This work simulated two scenarios, and the results reveal that the service negotiation solution, in addition to save energy and increase the network lifetime, allows the access point to be notified about the fire earlier and with more accuracy.

group in the Electrical Engineering Department of UFMG. She is also the leader of the MANNA research team.

THAIS REGINA DE MOURA BRAGA (thaisrb@dcc.ufmg.br) is a first year M.Sc. student at the Computer Science Department of UFMG. She received her B.Sc. degree in computer science from UFMG in 2004. Her research areas and main topics of interest include computer network management, wireless sensor networks (WSNs), distributed algorithms, and autonomic computing. Currently, she works in the SensorSim project, which deals with the development of a framework for wireless sensor network simulation. She has also participated in the Sensornet project (a research project funded by the CNPq to develop solutions for WSNs). She is also developing her M.Sc. dissertation on self-managed WSNs.

FABRICIO AGUIAR SILVA (fasilva@dcc.ufmg.br) is a first year M.Sc. student at the Department of Computer Science of Federal University of Minas Gerais (UFMG), Brazil. He received a B.Sc. in Computer Science from UFMG in 2004. Currently, he works at the SensorSim project, which deals with the development of a framework for wireless sensor networks simulation. He has also participated of Sensornet project (a research project funded by the CNPq — Brazilian Research Agency — to develop solutions for WSNs). His areas of interest and research include wireless sensor networks management, distributed network management and distributed systems.

HELEN PETERS DE ASSUNÇÃO (helen@dcc.ufmg.br) is a M.Sc. student in Electrical Engineering at the Federal University of Minas Gerais (UFMG), Brazil. She received a B.Sc. degree in Computer Science from UFMG in 2004. She has participated of Sensornet project (a research project funded by the CNPq — Brazilian Research Agency — to develop solutions for WSNs). Her research interests include Wireless Sensor Networks, Embedded Systems and Autonomic Networking.

JOSÉ MARCOS S. NOGUEIRA (jmarcos@dcc.ufmg.br) is an associate professor of Computer Science at UFMG — Federal University of Minas Gerais, Brazil. His areas of interest and research include computer networks, wireless sensor networks, telecommunications and network management, and software development. He received a B.S. degree in Electrical Engineering, a M.S. degree in Computer Science from UFMG in 1979, and a Ph.D. degree in Electrical Engineering from the University of Campinas, Brazil in 1985. He held a post-doctoral position at the University of British Columbia, Canada, 1988-1989, and currently is on a sabbatical year at the universities of Evry and Paris VI/LIP 6. He headed the Department of Computer Science at UFMG from 1998 to 2000. Currently, he heads the computer network group at UFMG. He was the technical coordinator of the System for the Integration of Supervision (SIS) Project where a complex and distributed system for the management of telecommunications networks was developed. He has served in various roles, including General Chair (1985) and TPC Chair (1999 and 2004) of the Brazilian Symposium on Computer Networks (SBRC), and General Chair of LANOMS 2001. He has been a TPC member in IEEE/IFIP NOMS (2000, 2002, 2004), IEEE/IFIP IM 2003, IEEE LANOMS (1999, 2001, 2003, 2005), IEEE/IFIP MMNS (2000, 2001, 2002, 2003), IPOM 2002, SBRC (from 1990 to 2005), and IEEE/IFIP DSOM (2003, 2004, 2005). He is a member of the Brazilian Computer Society (SBC) and Brazilian Telecommunications Society (SBrT). He also participates in the IEEE ComSoc CNOM interest group. Currently he is secretary of the IEEE ComSoc TCII. He has supervised a number of Ph.D. and Master's students, and publishes regularly in international conferences and journals.

ANTONIO A. F. LOUREIRO (loureiro@dcc.ufmg.br) holds a B.Sc. and a M.Sc. in Computer Science, both from the Federal University of Minas Gerais (UFMG), and a Ph.D. in Computer Science from the University of British Columbia, Canada. Currently he is an Associate Professor of Computer Science at UFMG. His main research areas are mobile Computing, distributed algorithms, and network management. He was one of the first researchers to work at Brazilian universities with sensor networks and mobile computing. He graduated 15 M.Sc. and 1 Ph.D. students in these two research areas. He is author of several papers in journals and conferences in the different areas of his research interests.