A Service Management Approach for Self-Healing Wireless Sensor Networks

Helen P. Assunção and Linnyer B. Ruiz

Electrical Engineering Department
Federal University of Minas Gerais
Av. Antônio Carlos, 6627
31279-010, Belo Horizonte, MG, Brazil
{helen, linnyer}@cpdee.ufmg.br

Abstract. Wireless Sensor Networks (WSNs) can provide three types of basic services: sensing, processing and disseminating. A shortcoming of any of these services can disturb the network goals. However, failures are not exceptions in WSNs, since problems as energy exhaustion, communication range loss or physical damages are usual incidents. Considering the need of managing WSNs in an efficient manner, improving information quality and availability, this work proposes the use of the IT Infrastructure Library (ITIL) and the autonomic computing paradigm in the design of self-healing WSNs. We also propose a service management approach for a self-managed network that dynamically adapts itself in order to maintain the service availability and promote the resources productivity. This approach aims to employ the self-healing service in WSNs, allowing them to discover, examine, diagnose and react to dysfunctions. Results show that service management applied to a self-healing WSN extend the longevity and availability of the network.

1 Introduction

A Wireless Sensor Network (WSN) is a distributed sensing tool and in this work, viewed as an Information Technology (IT) system. In the majority of cases, these networks are composed of hundreds of thousands of elements (sensor nodes), which are able to collect, process, disseminate and store data. The elements perceive the environment, monitor different parameters and collect data according to the application purpose.

The design of autonomic systems for WSNs must consider that this kind of network has particular characteristics that distinguish them from traditional networks, such as severe communication, energy and processing constraints, and deployment in remote and inhospitable environments without human intervention. For this reason, any hardware or software operation must be energy efficient, including self-management operations.

Due to the need of managing WSNs in an efficient manner, improving information quality and availability, moreover reducing costs, this work proposes the use of the IT Infrastructure Library (ITIL) and the autonomic computing paradigm in the design of an self-healing WSNs.

Autonomic computing [3] is a technology that defines systems that manage themselves and improve their operation without direct human intervention. ITIL [7] is a set of best practices to manage IT infrastructure and is the most widely accepted approach to IT service management in the world. Together these technologies incorporate an accepted model and independent of underlying structure, technology or architecture to the autonomic computing, meanwhile applies the ITIL processes to system with low level of human intervention.

As a study case, this work employs some of the ITIL concepts to provide the self-healing service, in case of WSN nodes failure. The system design involves events detection, notification and classification. Whenever this system detects improper operations or sensor nodes failures, the autonomic managers negotiate resources in an effort to recover the network automatically, keeping the services availability. The detected events can lead the system to conflict situations, which imply the need of learning skills to solve them. The proposed system works as a combination of strategies and policies that allows the description of the system behavior, the failures context and other events. These are information of major concern since they improve learning and planning tasks in a WSN.

This work is organized as follows. Section 2 presents Wireless Sensor Networks (WSNs) that self-manage themselves without direct human intervention. This section also presents the main management services for an autonomic WSN. Section 3 makes a brief presentation of the service support and service delivery processes of the Information Technology Infrastructure Library (ITIL). Section 4 presents one of the important contributions of this work - the definition of an Autonomic Service Management System, developed based on the autonomic computing and ITIL concepts. Section 5 presents a study case, in which the model proposed is instantiated for a WSN, considering the self-healing service. This section also presents the performed experiments as well as the obtained results. Section 7 aims to show some of the related work autonomic WSNs. Concluding the work, section 8 presents the final comments and future works.

2 Autonomic WSNs

WSNs assign a set of technological resources to the generation and employment of information. The network produces, processes and delivers its own data. The network is the user of its own services and can negotiate sensing, processing, storing and delivering services, according to the established goals.

In the majority of WSNs applications, network elements are deployed in remote areas where the maintenance and administration by technicians are impracticable. These elements, called sensor nodes, are designed with small dimensions and are expected to cost few dollars allowing the use of hundreds of nodes in different applications. Sensor nodes are composed by a computational unity, a wireless communication unity, a sensing unity (one or more sensors), a logic unity (software) and a power unity, which recharging is generally impracticable (in face of the great amount of nodes and the deployment in remote areas).

Sensor nodes perceive the environment, monitor different parameters and collect data according to the application purpose. In certain types of application, the network must collect, process and deliver data in validity period. In other types of applications, sensed data must be delivered to an application that correlates them and generates a report on the environment. Any missing or late information can influence the correlation results. For this reason, a failure in the element sensing, disseminating, storing and disseminating activities can disturb the network goals.

To design and develop energy efficient management systems in environments that impose severe restrictions is not a trivial task. Considering WSN characteristics, this work proposes that they should be autonomic, that is, they should self-manage themselves with the least human intervention.

An autonomic system is composed by interrelated autonomic elements. Each of these elements has managed resources, that is hardware or software that build the IT infrastructure, and autonomic managers that supervise and control these resources using an standard interface (touchpoint). The autonomic manager provides self-management services using monitoring, planning, analyzing and executing modules. Figure 1 presents the interaction between autonomic elements [2].

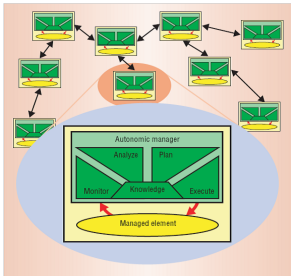


Fig. 1. Autonomic system.

Regarding to autonomic WSNs, the management tasks considers one or more situations perceived from the environment. These tasks should consider some aspects [11]:

- *Self-healing:* the management service that discovers, diagnoses and reacts to network disruptions. Self-healing components detect improper operations and failures and start corrective actions based in defined policies to recover the network or a node. The automatic recovering from damages improves the service availability.
- *Self-optimization:* the management service that maximizes the resource allocation and utilization, and guarantees optimal service quality, based on policies. The automation of complex tasks and the components adjustment in response to variable workloads allows the delivery of a high-level service.
- *Self-configuration:* the management service that changes configuration parameters to adapt itself dynamically under varying conditions and network states.

This service self-configures and reconfigures the network elements under varying and even unpredictable conditions. The network configuration must occur automatically, as well as dynamic adjusts to the current configuration to best handle changes in the environment.

- *Self-protection:* the management service that detects and protects the network against threats (internal or external, accidental or malicious). In case an attack happens, this service executes detection routines in order to reach security.
- *Self-service:* the management service that allows the provision of sensing, processing and dissemination services, anticipating resources and at the same time keeping the complexity hidden, in order to shrink the gap between business application and service goals.
- *Self-awareness:* the management service that allows the entity to know its environment and its activities context and act accordingly. It finds and generates rules to best interact with neighbors entities.
- *Self-knowledge:* the management service that qualifies an entity that knows itself. For example, an entity that governs itself should know its components, current state, capacity and all the connections with other entities. It needs to know the extension of its resources that can be lent and borrowed.
- *Self-maintain:* the management service that allows an entity to monitor its components and fine-tune itself to achieve pre-determined goals of an entity.

In this work, the autonomic WSN is composed of autonomic sensor nodes - the smallest part of the autonomic system. Four common functions, identified in the characteristics describe above, should be implemented in the autonomic sensor nodes: a function to collect the details it needs from the system; a function to analyze those details to determine if something needs to change; a function to create a plan, or sequence of actions, that specifies the necessary changes; and a function to perform those actions. These functions work together to provide the control loop functionality of an autonomic manager [2]:

Monitor. The monitor function encloses system (hardware, software) and environment monitoring in order to extract the behavior of these components. This function collects details from the managed resources, and aggregates, correlates and filters them into symptoms that can be analyzed. The details can include topology information, metrics, configuration, status, capacity and throughput.

Analyze. The analyze function provides mechanisms to observe and analyze situations in an effort to determine whether changes need to be implemented. This function can model complex behaviors to use prediction techniques allowing the autonomic managers to learn about the IT environment and predict future behaviors.

Plan. The plan function creates or selects a procedure to execute a desired change in the managed resource. A change plan, which represents a desired set of changes for the managed resource, is created and passed to the execute function.

Execute. The execute function provides the mechanism to schedule and perform the necessary changes to the system. This function is responsible to execute the change plan and to update the knowledge used by the autonomic manager.

This work uses some of the IT Infrastructure Library (ITIL) concepts to employ self-management services in autonomic WSNs. For this purpose the monitor, analyze, plan and execute functions are implemented for autonomic managers defined under the ITIL paradigm.

3 The IT Infrastructure Library

The IT Infrastructure Library (ITIL) has become the most widely accepted approach to IT service management. It provides a consistent set of best practices for IT service management, promoting a quality approach to achieving business effectiveness and efficiency in the use of information systems [1].

The IT Infrastructure Library, documented in approximately forty books, describes the main processes of IT service management. The two main areas of ITIL are the Service Support and Service Delivery. Together, these two areas consist of ten disciplines that are responsible for the provision and management of effective IT services.

The Service Support disciplines include the Incident Management, Problem Management, Changes Management, Release Management and Configuration Management [5]. The Configuration Management discipline provides information about the IT infrastructure and controls this infrastructure by monitoring and maintaining information about all the necessary resources to deliver services. The Incident Management recovers the service functioning as fast as possible. In addition to that, it minimizes the incident impact on business operations and guarantees the offer of the best quality of service and availability, according to the Service Level Agreement (SLA). The Problem Management stabilizes the IT services by minimizing incidents consequences by the removal of their root cause, preventing incidents, problems and recurring incidents. The Change Management coordinates and plans changes in an efficient manner, with the minimum risk to the existing infrastructure. The Release Management implements the proposed changes, guaranteeing the use of authorized, tested and correct software and hardware while implementing these changes.

The Service Delivery disciplines include the Availability Management, Continuity Management, Capacity Management, Financial Management and Service Level Management [6]. The Availability Management predicts, plans and manages the service availability, considering aspects of reliability, maintainability and redundancy. This goal is reached by determining business availability requirements and adjusting these requirements to the capacity of the IT infrastructure. The Continuity Management plans alternative configuration items in an effort to recover from problems. The Capacity Management identifies and specifies the client's demands, and then, translates these needs into resources, guaranteeing the services performance. The Financial Management identifies, computes and manages the cost of delivering IT services. The Service Level Management maintains and improves the quality of IT services, by monitoring and adjusting these services.

These disciplines interact to guarantee that the IT infrastructure delivers to the business a high quality service. In the next section, ITIL is presented as an integration solution to employ self-management services and functions (monitor, analyze, plan and execute) in autonomic WSNs.

4 Service Management System for Self Healing for WSNs

Considering that failures are not exceptions in WSNs, this work deals with the design of a self-management system for WSNs that detects and identifies failures, and proposes adjusts to the network infrastructure, in order to maintain the service availability.

The architecture of the proposed system is described as follows. Based on Service Level Agreements (SLA), the system generates service management policies to the network elements. These policies are delivered to the network nodes and stored in repositories, the knowledge bases. The autonomic managers, located in the sensor nodes, start the monitor, analyze, plan and execute functions, based on these information.

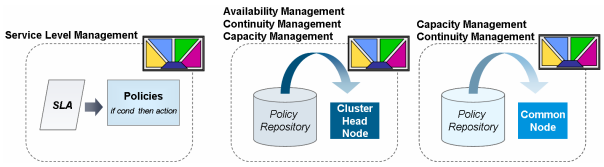


Fig. 2. The elements of an autonomic service management.

Each autonomic manager supervises some managed resources through the monitoring function. Their analyze function searches for policies stored in the knowledge base to determine if changes need to be implemented due to the occurrence of incidents. The plan function adjusts an existent solution to solve the problem according to its extension. Once a change plan is defined, the execute function applies it to the system and updates the knowledge source.

The proposed service self-management system consider the utilization of hierarchical WSNs, that is, networks in which sensor nodes are organized in clusters and each cluster has at least one leader. Autonomic managers and a policy repository, the knowledge base, are instantiated in the logic component of the sensor nodes, according to the nodes role in the network - common node or cluster head node role.

An autonomic manager, located outside the network, is responsible to map the Service Level Agreement (SLA) into policies for the network nodes, to monitor the service quality and availability, and if necessary, to renegotiate the SLA.

Autonomic managers that control cluster head nodes are responsible to guarantee that the service level is being attended inside the cluster and, otherwise, to adjust the network components to attend these levels

Common nodes autonomic managers are responsible to monitor their resources, optimize the nodes functioning, detect anomalous behavior, analyze events and adjust the nodes configuration in order to diminish problem risks. In case any problem occur, these managers will recover the network functioning as fast as possible. Common nodes include in their logic component not only autonomic managers and the knowledge source, but also touchpoints to interact with managed resources.

Some concepts of the ITIL Service Delivery area were employed in the definition of four autonomic managers with the purpose of creating a self-healing WSN, namely:

- Autonomic Service Level Manager: the autonomic manager that guarantees the fulfillment of agreed service levels and, eventually, redefines the SLA, using a manual manager or policies.
- Availability Autonomic Manager: the autonomic manager that plans and manages service availability through the monitoring of the IT service availability.
- Continuity Autonomic Manager: the autonomic manager that analyzes network risks identifying possible failures and creating a recover or risk reduction plan.
- Capacity Autonomic Manager: the autonomic manager that monitors nodes resources and identifies demands. In case of current or future insufficient capacity this manager is responsible to reallocate resources and anticipate new resources, what makes necessary the definition of a resource utilization model to determine whether the nodes are attending the defined requirements or not.

Each one of these managers employs concepts of Service Support disciplines to accomplish the monitor, analyze, plan and execute function, considering the self-healing service (see Figure 3).

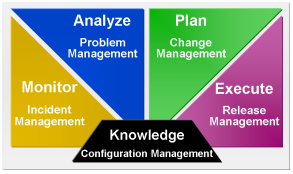


Fig. 3. Autonomic element.

The monitor function will detect events that are not part of the service normal operation and that can cause a disruption of the service or diminish its quality. The analysis function will perform the incident analysis, error diagnosis and root cause determination, and then determine if changes need to be performed triggering a Request for Change (RfC) if necessary. The plan function will propose changes to sensor nodes in order to solve network problems. The execute function will implement the proposed change plan in the network elements, and evaluate the change impact over the network.

Information utilized and generated by these four functions are stored in the nodes knowledge sources. The WSN autonomic managers are responsible to generate, manage and store information about the sensor nodes resources in the knowledge source.

The managers defined in this section will contribute to maintain the WSNs service availability.

5 Study Case

This section presents a study case utilizing the service management approach, proposed in this work, for a hierarchical WSN, in which incidents were instantiated as communication problems. The system must detect the root cause of the problems - in this study case problems are caused by traffic congestion and energy loss. After detecting improper operations or components problems the system will automatically recover from disruptions, maintaining service availability and continuity. Autonomic managers use residual energy data and production information as source of information to the monitor and analyze functions.

The knowledge bases are distributed between the network nodes. Common nodes store information about their own state. Cluster head nodes store not only these information but also network management information, once these nodes have more communication and storage capacity.

The main tasks defined to the autonomic managers located in common nodes are described bellow.

Monitor. The continuity manager detects message loss incidents. In order to detect if messages were lost, the nodes periodically check if they received an ACK message for each sensed data message sent. The capacity manager is responsible to monitor the sensor node residual energy. When the energy decays a determined threshold, an incident is detected and the node starts disseminating only high priority messages. Besides, this manager is able to detect abnormal production increase incidents, which occurs when some event of interest is sensed making nodes to increase their production.

Analyze. The error diagnose is accomplished using information stored in the sensor nodes knowledge sources. In case common nodes lose messages, the autonomic manager will diagnose if an abnormal increase in nodes production is the root cause of the incident (this increase makes nodes lose their messages since the packet queue is full).

Plan. After the autonomic manager detects that messages were lost and the production has increased, it will try to decrease the node's production, until messages are no longer lost. In order to control their production the manager proposes in the changes plan gradual increase of the sensing and dissemination interval. In case of a low energy problem the manager will propose the stop the node activities for some seconds.

Execute. The manager alters the sensing and disseminating intervals or puts the node out of service for a few seconds. While messages are being lost, the

manager keeps adjusting these parameters and evaluating their impact over the network.

The main tasks performed by each one of the autonomic managers located in the WSN cluster head nodes are described bellow.

Monitor. The detection of message loss is accomplished by the continuity manager using the same *ACK* mechanism of the common nodes. The capacity manager is responsible to monitor the sensor node residual energy. Cluster head node managers, also detect the network increase of production incident and low energy level in the cluster.

Analyze. In case cluster head nodes lose messages, these nodes analyze the knowledge source in order to diagnose if the failure is an abnormal production increase.

Plan. If the autonomic manager detected a low residual energy incident in its cluster, it proposes as change that the 20% of the cluster nodes with the smallest residual energy will stay out of service for a 5 seconds interval. A similar change is performed when the network has an abnormal production increase (which is characterized by a significant raise in the number bytes sent). The cluster head node chooses 20% of the cluster nodes with the highest production and put these nodes out of service for a 5 seconds interval. These plans aim to increase the network lifetime and deliver data rate.

Execute. The autonomic manager execute the change plan putting the chosen nodes out of service for a few time, and then evaluates the impact of the change over the network.

The knowledge source is updated whenever messages are received or configuration parameters are changed. Local information stored in the repository are: *id*, *energy*, *bytesSent*, *sens_interval*, *diss_interval*, *drop_number*, *last_drop_number*, *bytes_sent*, *bytes_dropped*, *bytes_dropped_second*, *bytes_second*, *last_bytes_second* e *priority*.

The cluster head nodes store in the knowledge source their local information and some replied information of each node of the cluster, namely: *id*, *energy*, textitdiss_interval, *bytes_sent*.

Each message received by the cluster head from an unknown node triggers the action of creating an entry for this node in the knowledge source (KS). After that, this repository will be updated by messages sent periodically from the common nodes with information about their repository entries. Besides, the cluster head knowledge source entries are modified when a sensed data message received, updating the received message rate from each node of the cluster.

The management application described in this section was simulated and the experiments are presented in the next section.

6 Experiments

This section presents the simulation scenarios and experiments simulated using the Network Simulator 2 (NS-2) tool and the MannaSim [15], which is a framework made of a set of base classes that extends NS-2 to simulate sensor networks.

The simulation scenarios were built with dimensions of 50 x 50m containing: four cluster head nodes and five common nodes per cluster. The cluster head nodes were positioned in a grid and the common nodes deployed randomly. An access point (AP) was positioned in the center of the scenario. Using this topology, two scenarios were simulated.

Scenario 1: The service management system for self-healing WSNs described in section5 was implemented.

Scenario 2: the network does not implements self-management functionalities that is, autonomic computing services were not implemented.

Network Configuration	**Simulations Configuration**
Cluster Head Nodes Number: 4;	*Simulation Time:* 155 seconds;
Common Nodes Number: 20;	*Number of Simulations:* 33;
Transport Protocol: UDP;	*Scenario Size:* 50 x 50m;
MAC Protocol: IEEE 802.11;	
Cluster Head Nodes Configuration	**Common Nodes Configuration**
Range: 250 metros;	*Range:* 40 metros;
Processing Consumption: 0.360W;	*Processing Consumption:* 0.024W;
Transmission Consumption: 0.6W;	*Transmission Consumption:* 0.036W;
Reception Consumption: 0.3W;	*Reception Consumption:* 0.024W;
Sensing Consumption: -	*Sensing Consumption:* 0.015W;
Disseminate Type: Programmed;	*Disseminate Type:* Programmed;
Sensing Type: -	*Sensing Type:* Programmed;
Battery Capacity: 100J;	*Battery Capacity:* 5J;
Bandwidth: 100kbps;	*Bandwidth:* 28.8kbps;

Table 1. Characterization of performed simulations.

The simulations characterization is presented in Table 1. The specifications of the two kinds of sensor nodes utilized in the simulations, common and leaders, where configured according to the ones presented by the real nodes Mica Motes [4] and WINS [14], respectively. Each scenario was executed 33 times and the results, presented in section 6.1, refers to the average of the obtained values.

The implemented application accomplishes the temperature monitoring using sensor nodes thermistors. The average sensed temperature is 25C with standard deviation of 5C. When the sensed temperature exceeds 28 c, the message that contains this data is considered a high priority message. Common nodes and cluster head nodes aggregates produced data and received messages, respectively, and disseminate the aggregated message periodically.

In order to simulate communication problems, at each 20 seconds, common nodes have their sensing and disseminating interval decreased from 0.01 and 1 seconds to 0.001 and 0.01 seconds, promoting an increase in network data production. As a consequence to that, messages are lost because of space loss at

the packet queue, that admit 10 messages in common nodes and 100 messages in cluster head nodes.

The network nodes try to detect incidents at every 1 second interval. In addition to that, common nodes send messages to update the cluster head knowledge base at each 1 second interval.

6.1 Results

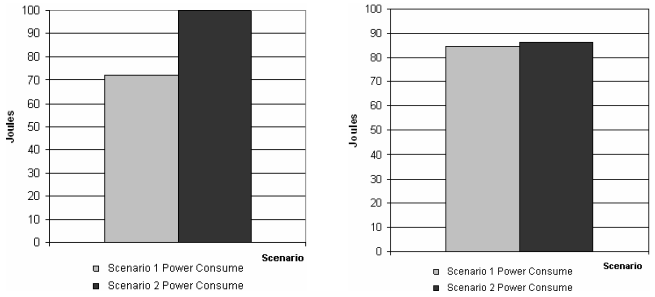
This section presents the simulation results and their evaluations.

Power Consumption

Energy is the resource responsible by the functioning of all sensor nodes modules. In case of nodes failure due to energy exhaustion, the network production is reduced in an irreversible manner.

Sensor nodes from scenario 2 get out of service after 75 seconds of simulation, supporting the communication problem event for a 25 seconds period. On the other hand, the common nodes of scenario 1, which implements the self-management system described in section 5 survive during the entire simulation, enduring the communication problem for a 60 seconds period, in which nodes detect incidents and adapt themselves in order to keep the service available.

The graphic of Figure 4(a) presents the power consumption in the common nodes for both simulation scenarios.



(a) Common nodes power consumption. (b) Cluster head nodes power consumption.

Fig. 4. Network Power Consumption.

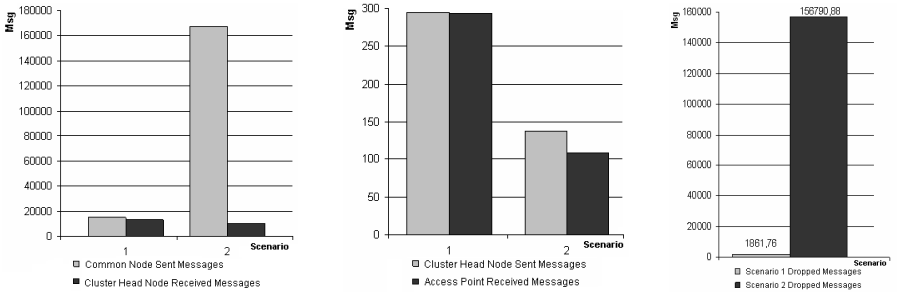
The power consumption of scenario 1 (see Figure 4(a)) is smaller than the power consumption in scenario 2, even though the network keeps functioning for an 80 seconds period larger than the scenario 2. This happened because nodes detect that messages are being lost and diminish their production. Besides, the fact that common nodes disseminate only priority information, in case of low energy level, promotes power saving.

The cluster head nodes from both scenarios presented a similar power consumption (see Figure 4(b)). Although the cluster head nodes from scenario 1 delivered more information, the size of the messages was smaller, making the power consumption comparable to the ones of scenario 2.

Sensed Data Flow

Figures5(a) and 5(b) presents the sensed data flow in the network. The amount of data sent by the common nodes of scenario 1 is 10,80 times greater than the one of the scenario 2 (see Figure 5(a)). However, the amount of data received by the cluster head nodes from scenario 1 is 1,3 times greater than the one of scenario 2. It means that the scenario 1 saves energy, since it delivers and drops less data because it tries to deliver messages when the network is not congested, and, even though, the number of messages delivered to the leader node is bigger if compared to the one of scenario 2. This demonstrates that the number of dropped messages in the network was diminished.

The amount of data sent by the cluster head nodes to the access point in scenario 1 is 2,16 times greater than the one in scenario 2 (see Figure 5(b)). This happens because the leader nodes decide to decrease their dissemination intervals, being able to deliver the received messages by the common nodes of the cluster. Even though, the scenario 1 presented only 0,40% of message loss, while the scenario 2 presented 21,07% of message loss.



(a) Common nodes sent messages and cluster head nodes received messages.

(b) Cluster head nodes sent messages and access point received messages.

(c) Network dropped Messages.

Fig. 5. Network sensed data flow and message drop.

The amount of dropped messages is presented in Figure 5(c). The scenario 2 drops number is 84,22 times greater the scenario 1, because it is not implemented any way of detecting communication problems nor either recovering from these problems.

Analyzing all the obtained results, the proposed solution has proved to be efficient in correcting communication problems and prolonging the network lifetime.

7 Related Work

It is notable the WSN area progress. However, in the scientific view, these networks present a great variety of new problems not studied yet or still incipient. This is the case when it is considered the WSNs as information technology systems, or even when it is intended to define these networks as autonomic systems.

For this reason, this work aims to be a contribution to the area when it deals with the challenge of applying the practice of ITIL processes and autonomic computing to WSNs. To the best of our knowledge, there is no directly related work to the use of ITIL in WSN, so this section presents some existing works on autonomic systems and autonomic WSNs.

An autonomic architecture to WSNs was proposed in [10]. The Autonomic-Oriented Architecture (AoA) was defined to support WSN self-organization. This architecture is allows a dynamic and intelligent control of networks built through intelligent software agents that offer reliability, quality of service, security, and mobility management.

The utilization of autonomic computing to perform symptom analysis is described in [8]. The main contribution of this work is the definition of a symptom model and its components (symptom artifacts and relationships). In [9] the author presents some IT scenarios that benefit from a symptoms-based autonomic computing architecture such as security, service support, service availability and service continuity.

The work [13] considers self-healing aspects in autonomic networks, in particular, techniques for events correlation and fault identification. The authors proposed an environment that performs problem determination and rule discovery for fault identification in telecommunication systems using alarm events.

Some autonomic functionalities were implemented for a WSN in [12]. This work defines policies to the accomplishment of the self-organization, self-awareness, self-knowledge services and service negotiation. The results showed that implementing autonomic functionalities and service negotiation benefits the energy saving and the quality of the information extracted from the network.

8 Conclusion

This work considers WSNs as IT tools or systems, since they produce, process, store and deliver data. As an IT system, the WSNs are based on the following components: sensor nodes hardware and software, network elements communication, and network produced information management. This work also proposes that WSNs should be designed as autonomic systems that implement different management services, such as self-organization, self-configuration, self-diagnosis and self-healing.

To the best of our knowledge, this is the first work in literature that considers the use of ITIL best practices to autonomic WSNs. The results from the scenarios, which implement the service management system for self-healing WSNs, show that this can be a great solution to manage failures, considering that these are not exceptions in this kind of network.

The self-healing system designed as study case deals with the specification of the monitor, analyze, plan and execute autonomic manager functions, which were modeled according to some of the ITIL practices.

The results show that the detection of improper operations and components failure in WSNs and the automatic recovering of problems promote a greater

availability of the service and the network longevity.

The autonomic WSNs promote the independence of human intervention in tasks of maintenance and management, reducing the communication costs, the response time to events and increasing significantly the network availability. The design and implementation of autonomic WSN represent a new research opportunity and, specially, when these networks are under the IT paradigm.

References

1. Hochstein, A., Zarnekow, R., and Brenner, W. (2005). Itil as common practice reference model for it service management: Formal assessment and implications for practice. *IEEE International Conference on e-Technology, e-Commerce and e-Service*.
2. IBM (2005). Autonomic computing white paper - An architectural blueprint for autonomic computing.
3. Kephart, J. O. and Chess, D. M. (2003). The vision of autonomic computing. *IEEE Computer*, 36(1):41–50.
4. MICA (2003). *MICA Wireless Measurement System*. Crossbow Technology Inc. Available at http://www.xbow.com/
5. Office of Government Commerce (2000). *ITIL Service Support*. The Stationery Office.
6. Office of Government Commerce (2001). *ITIL Service Delivery*. The Stationery Office.
7. Office of Government Commerce (2002). *Planning to Implement Service Management Manual*. The Stationery Office.
8. Perazolo, M. (2005a). Symptoms deep dive, part 1: Know thy symptoms, heal thyself. IBM Autonomic Computing DeveloperWorks.
9. Perazolo, M. (2005b). Symptoms deep dive, part 2: Cool things you can do with symptom. IBM Autonomic Computing DeveloperWorks.
10. Pujolle, G. and Chaouchi, H. (2005). An autonomic oriented architecture for wireless sensor networks. *Annals of Telecommunications - Sensor Networks*, 60(6/7):819–830.
11. Ruiz, L. B. (2003). *MANNA: A Management Architecture for Wireless Sensor Networks*. PhD thesis, Computer Science Department of the Federal University of Minas Gerais, Belo Horizonte, MG, Brazil.
12. Ruiz, L. B., Braga, T. R. M., Silva, F., Assunção, H. P., Nogueira, J. M. S., and Loureiro, A. A. F. (2005). On the design of a self-managed wireless sensor network. *IEEE Communications Magazine*, 43(8):95–102.
13. Sterritt, R. (2004). Autonomic Networks: engineering the self-healing property. In *Journal of Advanced Engineering Informatics, Engineering Applications of Artificial Intelligence*, volume 17, pages 727–739 187. Elsevier Publishers.
14. WINS (2002). Wireless Integrated Network Sensors (WINS). Available at http://www.janet.ucla.edu/WINS/.
15. Lopes, C. E. R., Melo, J. C., Assunção, H. P., Braga, T. R. M., Silva, F. A., Ruiz, L. B., Loureiro, A. A. F. and Nogueira, J. M. S, (2006). MannaSim: Simulando Redes de Sensores Sem Fio. *24th Brazilian Symposium on Computer Networks (SBRC'06)*.

This article was processed using the LaTeX macro package with LLNCS style