International Journal of Emerging Trends in Engineering and Basic Sciences (IJEEBS)
ISSN (Online) 2349-6967
Volume 2, Issue 3(May-June 2015), PP 96-101

Self-Healing Wireless Sensor Networks

Parag Puranik[1], Sagar Motdhare[2]
[1]*(Department of Electronics and Communication Engineering, PIGCE/RTMNU, Nagpur,India)*
[2]*(Department of Electronics Engineering, PIGCE/RTMNU, Nagpur,India)*

Abstract :- Nowadays wireless sensor networks have found their way into a wide variety of applications and systems with vastly varying requirements and characteristics, but all of them have a common element: faults are a normal fact and not isolated events as in traditional networks. Thus, in order to guarantee the network quality of service, it is essential for the sensor network to be able to detect and heal failures. In this work a failure detection scheme and a service management approach using the autonomic computing paradigm and some concepts of the IT Infrastructure Library (ITIL) will be evaluated. The presented approach aims to employ self-healing services, allowing them to discover, examine, diagnose and react to malfunctions

Keywords :- Autonomic computing, fault tolerance, net-work architecture and design, network management, self-healing, wireless sensor networks

I. INTRODUCTION

Today we are at the beginning of that *ubiquitous computing* era. The design of micro power wireless sensor systems has already gained increasing importance for a variety of civil and military applications. Recent advances in micro-electro-mechanical systems (MEMS) technology and its associated interfaces, signal processing, and wireless communications, have enabled the development of low-cost, low-power, multifunctional sensor nodes that are small in size and communicate in short distances. These tiny sensor nodes, which consist of sensing, data processing, and communicating components, leverage the idea of sensor networks based on collaborative effort of a large number of nodes and so the focus has shifted away from limited macro sensors communicating with base stations.

While individual micro sensor nodes are not as accurate as their macro sensor counterparts, the networking of a large number of nodes enables high quality sensing networks with the additional advantages of easy deployment and fault tolerance. These characteristics make micro sensors ideal for deployment in otherwise inaccessible environments, where maintenance would be inconvenient or impossible and represent a significant improvement over traditional sensors, which are deployed in the following two ways:

- Sensors can be positioned far from the actual phenomenon, i.e., something known by sense perception. In this approach, large sensors that use some complex techniques to distinguish the targets from environ-mental noise are required.
- Several sensors that perform only sensing can be deployed. The positions of the sensors and communications topology are carefully engineered. They transmit time series of the sensed phenomenon to the central nodes where computations are performed and data are fused.

A common sensor network is composed of a large number of sensor nodes cases, these networks, which are densely deployed either inside the phenomenon or very close to it.

The position of sensor nodes need not be engineered or pre-determined. This allows random deployment in inaccessible terrains or disaster relief operations. On the other hand, this also means that sensor network protocols and algorithms must possess self-organizing capabilities because deploying and maintaining the nodes must remain inexpensive – manually configuring large networks of small devices is impractical.

The nodes are able to collect, process, disseminate and store data. They perceive the environment, monitor different parameters and collect data according to the application purpose. Another unique feature of sensor networks is the cooperative effort of sensor nodes. Sensor nodes are fitted with an on-board processor. Instead of sending the raw data to the nodes responsible for the fusion, sensor nodes use their processing abilities to locally carry out simple computations and transmit only the required and partially processed data. The reason for this is that computation is much cheaper than communication in regard to the most critical resource, the energy. Each transferred bit costs as much energy as about 1,000 instructions, thus, wireless sensor networks process data within the network wherever possible.

A wireless sensor network has applications in environmental and habitat monitoring, precision agriculture, indoor climate control, surveillance, treaty verification, intelligent alarms, and medical diagnostics. The most dramatic applications involve monitoring complex interactions, including wildlife habitats, disaster management, emergency response, ubiquitous computing environments, asset tracking, health-care, and manufacturing process flow.

Some of these applications require a large number of devices making traditional methods of sensor networking impractical due to the high demand on cable installations.

To be able to manage wireless sensor networks in an efficient manner, Assunção *et al.* proposed the use of the IT Infrastructure Library (ITIL) and the autonomic computing paradigm.

II. AUTONOMIC WIRELESS SENSOR NETWORKS

In the majority of cases the network elements, called sensor nodes, of a wireless sensor network are deployed in remote areas where maintenance and administration by technicians are impracticable. The form factor of a single sensor node can vary, depending on the actual need of the application, from the size of a shoe box (e.g. a weather station) to a microscopically small particle (e.g. for military applications where sensor nodes should be almost invisible). Similarly, the cost of a single device may vary from hundreds of Euros (for networks of very few, but powerful nodes) to a few cents (for large-scale networks made up of very simple nodes). Each device is composed by a computational unit, a wireless communication unit, a sensing unit (one or more sensors), a logic unit (software) and a power unit. Recharging or replacing the battery is generally impracticable, since there are thousands of nodes in potential inaccessible environments. Depending on the application, the required lifetime of a sensor network may range from some hours to several years and has a high impact on the required degree of energy efficiency and robustness of the nodes.

Sensor nodes observe the environment, monitor different parameters and collect data according to the application purpose. In some applications, the network must collect, process, and deliver the data continuously and in real-time, in other types of applications the data is delivered to the observer only when a certain event occurs. Any missing or late information can influence the interpretation of the data and therefore a failure in sensing, processing, or delivery can disturb the network goals.

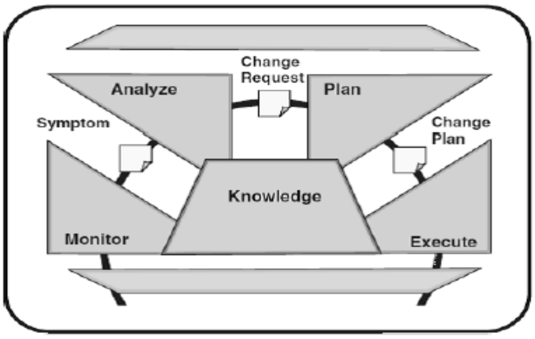

Fig 1 : Interaction between Autonomic Elements

The design and development of energy efficient systems in environments that impose severe restrictions is not a tri-vial task. Considering the given characteristics, the system should be as autonomic as possible, that is, the wireless sensor network should manage itself with the least or no human intervention.

An autonomic system is composed of interrelated autonomic elements. Each of these elements has managed hard ware or software resources that build the IT infrastructure and autonomic managers that supervise and control these re-sources. The autonomic manager provides self-management services using monitoring, planning, analyzing and executing modules. Fig.1 presents the interaction between the autonomic elements.

Regarding to autonomic wireless sensor networks, the management tasks should consider some of the following aspects:

Self-healing: discover, diagnose, and react to network disruptions. Self-healing components can detect system malfunctions or failures and start corrective actions based on defined policies to recover the network or a node. The automatic recovering from damages improves the service availability.

Self-optimization: monitor and tune resources automatically. The management services that maximizes the resource allocation and utilization, and guarantees optimal service quality, based on policies. The tuning actions could mean reallocating resources – such as in response to dynamically changing workloads – to improve overall utilization, or ensuring that particular business transactions can be completed in a timely fashion. The automation of complex tasks and the components adjustment in response to variable workloads allows the delivery of a high-level service.

Self-configuration: change configuration parameters to adapt dynamically under varying conditions and net-work states. This management 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: anticipate, detect, identify and protect against threats (internal or external, accidental or malicious) from anywhere. In case an attack happens, this service executes detection routines in order to reach security.

Self-service: allow 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: allow the entity to know its environment and its activities context and act accordingly. It finds and generates rules to best interact with neighbor entities.

Self-knowledge: the management service that qualifies an entity to know 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: allow an entity to monitor its components and fine-tune itself to achieve pre-determined goals.

According to the characteristics described above, four common functions should be implemented in the autonomic sensor nodes: a function to collect the details it needs to know from the system, a function to analyze those retrieved details to determine if something is wrong and needs to change, a function to create a plan, or sequence of actions, that specifies the necessary changes, and a function to per-form those actions. These functions work together to pro-vide the control loop functionality of an autonomic manager:

Monitor: The monitor function provides the mechanisms that collect, aggregate, filter and report details that can be analyzed and collected from a managed resource. The details can include topology information, metrics, configuration, status, capacity, and throughput.

Analyze: The analyze function provides mechanisms to correlate, observe, and analyze complex 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. The planning mechanism uses policy information to guide its work.

Execute: The execute function provides the mechanisms that controls the execution of a plan with considerations for dynamic updates. This function is responsible to execute the change plan and to update the knowledge used by the autonomic manager.

III. FAILURE DETECTION AND FAULT MANAGEMENT

Sensor nodes have strong hardware and software restrictions in terms of processing power, memory capability, power supply, and communication throughput. The power supply is the most critical restriction, given that it is typically not rechargeable. For this reason faults are likely to occur frequently and will not be isolated events. Besides, large-scale deployment of cheap individual nodes means that node failures from fabrication defects will not be un-common.

In military applications, where these networks are deployed in open spaces or enemy territories, adversaries can manipulate the environment (so as disrupt communication, for example by jamming), but have also physical access to the nodes. At the same time, ad-hoc wireless communication by radio frequency means that adversaries can easily put themselves in the networks and disrupt infrastructure functions (such as routing) that are performed by the individual nodes themselves. Finally, the sensor nodes are exposed to natural phenomena like rain, fire, or even falls of trees since they are commonly used to monitor external environments. Therefore failure detection and fault management plays a crucial role in wireless sensor networks. If, in addition to detecting a failure, the management application can also determine the reasons of the failure and distinguish between malicious and non-malicious origins, it can trigger security management services or, if it is an accidental or natural failure, activate “backup nodes”.

In applications interested in the conditions of the environment at all times sensor nodes will be programmed to sense and send back their measurements at regular intervals or continuously. These networks are called *programmed* and *continuous*, respectively. Other applications just need data when some “special” events occur; the networks are then called *event-driven* networks. On the other hand, when the network is able to answer to queries of the observers, it has the *on demand* property.

Configuring the network as event-driven is an attractive option for a large class of applications since it typically sends far fewer messages. This results into a significant energy saving, since message transmissions are much more energy-intensive when compared to sensing and processing. For instance, if the application is temperature monitoring, it could be possible just to report data when the temperature of the area being monitored goes above or below certain thresholds.

IV. SERVICE MANAGEMENT SYSTEM FOR SELF-HEALING

To extend this model of failure detection and fault management, Assunção *et al.* proposed in their work the usage of some of the IT Infrastructure Library (ITIL) concepts and proposed a system whose architecture is described as follows:

An autonomic manager, located outside the network, is responsible to map the Service Level Agreement (SLA) into so called “policies” for the network nodes, to monitor the service quality and availability and, if necessary, to renegotiate the SLAs. These policies are delivered to the net-work nodes and stored in repositories, the knowledge bases. The autonomic managers, located in the sensor nodes, then start the monitor, analyze, plan and execute functions, based on this information.

Autonomic managers on cluster-head nodes guarantee that the service level is being attended inside the cluster and 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 the risk of faults. In case any problem occurs, these managers will recover the network operation as fast as possible.

Some concepts of the ITIL Service Delivery area were employed by Assunção *et al.* in the definition of four autonomic managers with the purpose of creating a self-healing wireless sensor networks, 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.

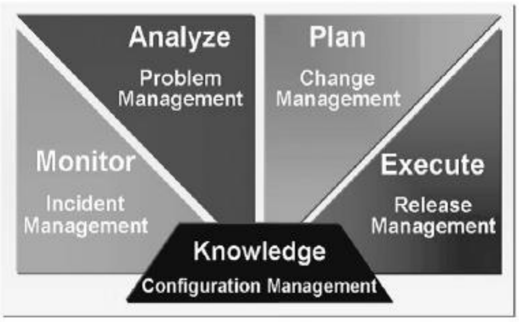

Fig 2: Autonomic Element

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 Fig. 2).

V. CONCLUSION

Building and deploying networks, especially in environments where there will be tens of thousands of network elements with particular features, is very complex. The task becomes even worse due to the physical restrictions of the sensor nodes, in particular energy and bandwidth restrictions and in some applications even the environment is inaccessible. Nevertheless it is possible to operate a wireless sensor network in such circumstances. The autonomic computing approach is one of the possible solutions, because it helps to keep the network independent of human interventions. As the results show the detection of improper operations and components failure in wireless sensor networks and the automatic recovering of problems promote a greater availability of the service and the network longevity. Analyzing all the obtained results, the proposed solution has proved to be efficient in correcting communication problems and pro-longing the network lifetime even if – at least in an event-driven network – there will be an overhead that is accepted.

REFERENCES

[1] Gordon E. More, “Cramming More Components onto Integrated Circuits” *Electronics*, volume 38, number 8, April 1965.

[2] J. Kahn, R.H. Katz, and K. Pister, “Emerging Challenges: Mobile Networking for „Smart Dust,‟” *J. Comm. Networks*, pp. 188-196, Sept. 2000.

[3] C. Intanagonwiwat, R. Govindan, D. Estrin, “Directed diffusion: a scalable and robust communication paradigm for sensor networks”, Proceedings of the *ACM Mobi-Com’00*, Boston, MA, 2000, pp. 56–67.

[4] Bernhard Rinner, “Context-Aware Computing”, course at Graz University of Technology, 8010 Graz, Austria, 2006. Available: http://www.iti.tugraz.at/en/teaching/echtzeit_ki/index.html

[5] Helen P. Assunção, Linnyer B. Ruiz, and Antônio A. Loureiro, *A Service Management Approach for Self-healing Wireless Sensor Networks*, LNCS 4195, pp. 215-228, 2006,

[6] *Planning to Implement Service Management Manual. The Stationary Office,* Office of Government Commerce, 2002.

[7] IBM (2005), *Autonomic computing white paper – An architectural blueprint for autonomic computing,* IBM Corp., 2005.

[8] Linnyer Beatrys Ruiz, Isabela G. Siqueira, Leonardo B. e Oliveira, Hao Chi Wong, José Marcos S. Nogueira, Antonio A. F. Loureiro, “Fault Management in Event-Driven Wireless Sensor Networks”, Proceedings of the 7th ACM international symposium on Modeling, analysis and simulation of wireless and mobile systems MSWiM‟04, Venice, Italy, 2004, pp. 149-156.

[9] Kay Römer and Friedemann Mattern, “The Design Space of Wireless Sensor Networks”, *IEEE Wireless Communications Magazine*, volume 11, issue 6, pp. 54-61, Dec. 2004.

[10] Linnyer Beatrys Ruiz, Jose Marcos Nogueira, and Antonio A. F. Loureiro, “MANNA: A Management Architecture for Wireless Sensor Networks”, *IEEE Wireless Communications Magazine*, volume 41, issue 2, pp. 116-125, Feb. 2003.

[11] *MICA Wireless Measurement System*, Crossbow Technology Inc., 2003. Available at http://www.xbow.com/

[12] WINS (2002). *Wireless Integrated Network Sensors (WINS),* Department of Electrical Engineering, UCLA, Los Angel