On the self adaptiveness in Wireless Sensor Networks

Mikhail Afanasov
Politecnico di Milano
afanasov@elet.polimi.it

Sonda Bousnina
Politecnico di Milano
sonda.bousnina@ieee.org

Giuseppe Massari
Politecnico di Milano
giuseppe.massari@polimi.it

Vincenzo Sciancalepore
Politecnico di Milano
vincenzo.sciancalepore@imdea.org

***Abstract*—This work aims at providing a report, showing some noticeable example of self-configuration (or self-adaptation) approaches in the context of Wireless Sensor Networks (WSNs). Although this cannot be considered an exhaustive survey, we are confident about the fact that this work includes enough information to motivate and justify the exploitation of self-adaptiveness techniques in the design of such systems.**

I. Introduction

Wireless Sensor Network (WSN) is a set of tiny distributed wireless nodes that have to collaborate and cooperate on a common distributed application to perform tasks specified by a user (Fig. 1). Recently, the development of WSNs has received an increasing interest due to the fast technological advances in the fields of sensor technology, low power microelectronics, and low energy wireless communications. These networks are currently used in wide range of applications in the scientific, medical, commercial, and military domains, like for instance home automation, environment monitoring, industrial control, surveillance, security, healthcare, etc.

In this context, WSNs are intimately tied to and inherently dependent on an environment they operate in. It leads to a necessity to adapt to an unpredicted environmental dynamics. Adaptation at run-time provides more flexibility in such software behavior as energy management, network protocols, architecture reconfiguration and error handling. In fact, the application of self-organization concept to wireless sensor networks is seen as a key driver for improving the operation and maintenance of these networks. Thus, the self-organization can help reducing the cost of installation and management by simplifying some operational tasks through automated mechanisms pointing the self-configuration, self-optimization and self-healing [1]. Regarding self-configuration, it is triggered by incidental events for instance to add a new site or to introduce a new service or a new network feature. In that case, these lasts can require a re-configuration of a number of radio parameters or several resource management algorithms. For the self-optimization in WSNs, intelligent methods are applied to the processed measurements to derive an updated set of the radio parameters or the resource management parameters [1]. Sensor nodes and wireless links are subject to different errors in function of the size of perturbation in time, space, and energy. So applying self-healing methods can resolve these problems and the loss of coverage or capacity induced by such events [1]. In this regard, WSNs increasingly need self-organization, self-configuration and self-adaptation to changing conditions to ease management and operation [2].

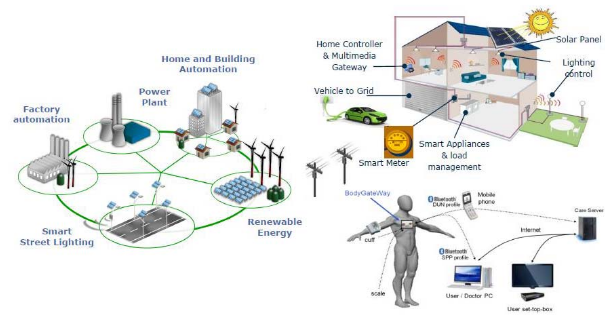

Fig. 1. Examples of wireless sensor networks.

In this respect, fits the present work. The remainder of this report is organized as follows. In section II, we present some related works in a sort of a state of the art about self configuration of WSNs. This section is subdivided in three parts describing respectively the concept of self configuration related to the WSNs architecture, the applications and services in these networks, and the systems and networking in these lasts. Overall evaluations about the works are discussed in Section III. Finally, Section IV concludes the paper.

II. State of the Art

In this section we group a set of relevant works focusing on the exploitation of self-adaptive and self-configuring approaches in WSN. We identified three perspectives, according to the targeted objective they face with. The *architecture* perspective looks at the WSN from the point of view of the topology, by adapting it to network dynamics, i.e. link failures, sensor nodes discovering and region covering. The *application* perspective copes with problems related to programming models, i.e. to deploy the application on a distributed system, update it, adapt the behavior of the software to changing environments. The *system* perspective aims at optimizing the resource usage, since WSN are very resource-constrained. We can perform energy management, taking into account QoS, by adapting network protocols, transmission power and duty-cycle efficiency enhancement.

A. Architecture

There is a significant part of work devoted to a self-adaptivity on an architectural level. Research in this area is focused mainly on such properties of the network as: topology, positioning of nodes, fault-tolerance and scalability. Unlike the other levels, architectural level implies a set of communicating and collaborating nodes.

An architecture for building self-configurable systems: Subramanian et al. [3] focused on the self-organizing architecture for WSNs and proposed components necessary for building such an architecture. The latter implies following components:

- *Specialized sensors* for monitoring different physical entities.
- *Routing sensors* provide a data dissipation and fault tolerance of the network.
- *Aggregator nodes* combine routing and sensor functionality to provide a better network flexibility.
- *Sink nodes* have a high storage capacity, store and process received data.

All the components mentioned above are sufficient for building a wide range of applications, where infrastructure consists of addressing, routing, broadcasting and multicasting mechanisms.

Given the proposed architectural components, author also propose four steps, which should be performed by the network to self-organize:

- *Discovery phase.* Each node discovers its neighbors.
- *Organizational phase.* Nodes are organizing groups, allocate addresses, build routing table and construct broadcast tree and graph spanning all nodes.
- *Maintenance phase.* Each node keeps track of its energy, constantly updates routing table, broadcast trees and graphs, sends *I am alive* message and routing table to its neighbors.
- *Self-Reorganization phase.* If node detects the failure of its neighbor, it updates its routing table, or starts *Discovery phase* in case of the failure of all of the neighbors.

The analysis of the approach shows that the hierarchy of the network is strictly balanced; the complexity of the routing is $O(\log n)$; the network is extremely tolerant to either node or link failures; the uniqueness property is guaranteed by the presence of a hierarchy; specialized sensors are allowed to be mobile. There are several weaknesses of the approach, though. Thereby, the approach is not optimized for extremely dynamic systems, when the network changes badly or very fast. On the other hand, the required protocol for such networks is not discussed. Despite the specialized sensors can be mobile, they can not move beyond the reach of routers. The latter, however are considered static.

The approach is very well fitted for static WSN, such as one for collecting meteorological data. But it is absolutely not applicable for systems with high dynamics such as wildlife monitoring [4], since the network changes badly and rapidly.

ASCENT: Adaptive Self-Configuring Sensor Networks Topologies: In [5] the authors provide an other self-configuring approach in order to deploy micro-sensor for a wide range of environmental monitoring applications. They assumed a very-dense scenario where is extremely necessary to find a trade-off between the coverage and the interferences due to the large number of involved sensors. The real scenario given by that work is a habitat monitoring sensor network that is deployed in a remote forest, where the sensors are dropped by a plane. Different aspects are taken into account and then analysed: power consumption as well as the distributed sensing task. The work seems to be very interesting for two different reasons: first, adaptive techniques implied permit applications to configure the underlying topology based on their needs by trying to save energy and extend the network lifetime, second, the self-adaptive approach is based on the operating conditions measured locally. Specifically, the authors identify two kinds of sensors, namely nodes, in the network: *i) active* and *ii)* passive. The active nodes stay awake all the time and perform routing procedure, while the passive nodes listen the channel and periodically check if they should turn into active mode. The active nodes will be in charge of producing messages (sources) or just disseminating them (sink). In the case of low channel condition, the sink nodes could send an *help message* to other passive node in order to activate them. The self-configuring process starts by turning on randomly some nodes in the network. Such nodes enter firstly in a test mode, where they can exchange data and routing control messages, so that after a prefixed time can be switched to an active mode. If there are too many neighbor nodes active, according to a fixed parameter, the node will be switched off. Afterwards, if the number of active neighbor nodes will be less than a prefixed parameter and the data loss rate greater than a predefined threshold, it will be re-activated. Note that the performance of the system depend on the value of the parameters above, e.g., loss threshold and neighbor threshold.

The work provides a good validation for the parameters provided, by analyzing and comparing the network performance in terms of energy savings and network capacity. The gain of power saving is a factor of 3 better in some dense cases.

B. Application

From an application perspective the challenges introduced by WSN are basically related to their distributed architecture, the scarceness of resources of the single sensor node, and the reliability issues due to sensor nodes disappearing for unpredictable faults, communication noises or battery discharging. This requires the development of ad-hoc programming models that could fit in the architectural view of the WSN, and that allow the application to react and adapt to environmental changes. This section introduces some approaches addressing this kind of problem.

COPAL-ML: A Macro Language for Rapid Development of Context-Aware Applications in Wireless Sensor Networks: Sehic et. al [6] propose a Java based API for developing self-adaptive applications for WSNs. By using this framework developer can specify the way the contextual data will be collected, processed and used by the application. COPAL-ML implies such components as:

- *Context type* specifies the format of the data provided by a WSN.
- *Publisher* periodically publishes an information about a specific context type.
- *Listener* handles an event fired by the publisher and receives a context type as an argument.
- *Processor* handles the received data.

Along with possible scenario, authors also proposed several processing patterns, which can be used as solutions for effective self-adaptive application. Despite it is very promising way towards abstraction of WSN on the high level, the framework is Java-based. Thus, it is applicable only for sensors with Java-machine built in. it is also can be used on a computer, but in this case the communication between the framework and the node remains unclear.

Supporting Lightweight Adaptations in Context-aware Wireless Sensor Networks: A common problem with WSN is to deploy an application, or update it, when the sensor network features a lot of nodes spread over a wide or region. As well, reprogramming a whole WSN deployed in a inaccessible region, represents a cumbersome, and often unfeasible, activity. Thus, the challenge is to support these operations in a feasible way, at the price of the lowest possible overhead.
Apart from this issues, a typical requirement of applications like *environmental monitoring* is to adapt the behavior of the nodes to the dynamics introduced by events, as for instance the presence or people moving around or not.
The work of Taherkordi et *al.*[7], introduces the `WiSeKit` distributed middleware, along with the `ReWiSe` software component model, as a solution to support a lightweight behavioral adaptation of sensor networks. The `WiSeKit` middleware exposes the following services

- *Local Reasoning* to update the values of components parameters based on a local adaptation policy.
- *Adaptation Proxy* to receive adaptation request from cluster head.
- *Component Repository* to temporarily store a new components image.
- *Component Reconfigurator* to load, reload or remove a running component.

The `ReWiSe` component model considers the application as an integration of separate components. A component can be dynamically replaced, Reconfiguration in WSN can be a very expensive operation, since it requires to transfer code to sensor nodes, to save and resume of state information, and eventually restart the sensor node. The main costs are paid in terms of *energy consumption* and are proportional to the size of the code images to transfer. For this reason, the goal of this work has been to develop a software component model that i) enables the possibility of a fine-grained reconfiguration of the application, ii) reduce the overhead due to state saving. The framework has been implemented on top of the `Contiki` operating system. Preliminary experiments reported a decrease of energy consumption of about 75%, compared to a common software component model.

C. System

The system level requires a lot of coordination between distinct sensors in the network. Particularly, several works leverage the WSN capabilities in order to reduce power consumption, make the MAC layer more flexible, and perform routing in a very efficient way. We take into consideration some of them in order to cover the main categories of system level improvements in the WSN.

Proactive Reconfiguration of Wireless Sensor Networks: Steine at *al.* in [8] leverage on the exploitation of a design-time exploration, to identify a set of operating modes. At run-time the WSN can adapt its behavior to environmental conditions and QoS fluctuations, by reconfiguring itself into the most suitable operating mode. The paper defines an *operating mode* as a set of values assigned to some controllable parameters of the network protocols. The nodes can dynamically change, at run-time, their operating mode according to specific observable events, that potentially affect the QoS of the WSN. The events are detected using sensors and/or the current time of the day. This figures out a *proactive* approach in the self-adaptive behavior. Concerning the network parameters, it is worth to distinguish between *local* and *global* parameters. In the former case, the single node can independently reconfigure itself without affecting the others. For instance, the tuning of the transmission power. In the latter, changing a global parameter necessarily requires a synchronization step, in order to maintain the proper functioning of the WSN. Example of global parameters are the TDMA slot-size and the sleep-time of the nodes. Thus, whenever a node detect an event, it can immediately adapt its local parameters or notify the other nodes about the need of change the current global mode. The authors have demonstrated the validity of the approach by testing it in a ”cow-health” and a ”office” monitoring scenarios. The evaluation metrics considered are the average delivery ratio and the average power consumption. In both the scenarios it has been experienced a significant reduction of power consumption, comparing the approach to a single configuration (worst-case) not adaptive design. This, keeping the packets delivery ratio close the values related to the worst-case based design.

pTunes: Runtime Parameter Adaptation for Low-power MAC Protocols: The work of Zimmerling et al. [9], is focused on an adaptation of system level – MAC protocol. The proposed framework – pTunes – allows to adjust the parameters of protocols to adapt to link, topology, and traffic dynamics. To this end, authors analyze the vital parameters – both protocol dependent and protocol independent – of the network, such as: reliability, latency and life-time. The framework also meets such requirements as: minimum disruption, timeliness, consistency and energy efficiency. Please note that pTunes deals exclusively with network protocol adaptation. Although the ASCENT[5] approach also touches network protocol parameters, it is more focused on network topology. Evaluation section of the work shows that the framework enhance the network lifetime, effectively controls the traffic and keeps the QoS even when link quality is very low. The approach, however, is centralized, thus it is impossible to perform per-node adjustment. It is also unclear, how much protocols are supported by the framework.

Use of self-adaptive methodology in wireless sensor networks for reducing energy consumption: From an energy saving perspective, interesting approaches have been proposed in the literature. One of them just exploits the idea that the energy used for communication in a sensor network can be improved by decreasing the communication range. The transmission energy is proportional to the distance between the transmitters. Therefore, in [10] they propose a self-adaptive methodology which reduce in an optimal fashion the distances between the main nodes in the network (BS) and the other sensors communicating with it as well as optimizes the number of active sensors by keeping the redundancy at good levels. The authors just divide the approach in two phases: the former is to identify the right location for the main sensor, which aims at minimizing the distances with the other sensor nodes, the latter is to switch off the nodes which are useless for covering the area. Additionally, since the nodes communicate by broadcasting messages, they suggested a smart solution which just enables in turn the active node to sense the channel in order to cope with the collision problem. The results are pretty good in terms of power saving. Interestingly, the proposed method not only reduces overall energy consumption of the network but also the lifetime of the nodes is increased significantly.

III. Evaluations

Despite there are a lot of work in the area of self-configurable Wireless Sensor Networks, the solutions are problem-specific, and there is no holistic approach for developing self-adaptive WSNs.

In this report, we classified the state of the art approaches according to three perspectives. Although, works are not strictly bound to a single perspective, like shown in [5].

The development of such systems can be a long and expensive process, since the developer should take care of all possible environmental conditions the system can experience. WSNs are usually built on very resource-restricted platforms, which makes it necessary to predefine all possible behaviour variations, since the adaptation to unpredicted situations is very expensive in terms of energy and memory consumption.

Self-adaptation is also expensive in terms of performance, since the system becomes less responsive during the analysis of the environment. Indeed, there also could be transitional states, where in the environmental conditions are changed and the old behaviour is no more suitable, but the new behaviour is not calculated or applied yet.

Thus, we can consider as costs of self-adaptivity for WSNs the reduction of both service time and responsiveness of the system, since additional calculations consume more energy and time. Therefore, self-adaptation for WSNs becomes an hard choice for such systems where dynamic changes do not frequently occur.

On the other hand, self-adaptation schemes lead to relevant advantages in terms of performance. The most important benefit, targeted by the majority of the works, is the energy saving. Specifically, a smart usage of resources, such as switching off some nodes, reducing the transmitting power, and so on, provides an useful maximization of the battery lifetime by improving the energy consumption.

Moreover, the self-adaptation techniques help to perform a more efficient coverage of the area in the Wireless Sensor Networks. Indeed, the self-adaptation provides better performance by optimizing the placement of nodes. At the same time, this leads to a more reliable system by assuring fault-tolerance. Interestingly, the nodes which are not involved in the coverage process, may be activated for supplying the sinks.

Finally, we can state that the most important feature introduced with the self-adaptive wireless sensor networks is the flexibility, as well as the ability to proficiently react to the environment changes. In this sense, the self-adaptation paradigm plays a key-role in terms of modifying system behaviours to fulfill the network requirements, by monitoring the current performance of the system.

IV. Conclusions

The presented paper shows some noticeable example of an interesting approach, called self-configuration (or self-adaptation) in the context of Wireless Sensor Networks (WSNs). The survey collects different proposals for any of the above-mentioned categories: the architecture layer, the application layer and the system layer. The work sheds light on what we consider a hot topic in the wireless sensor field, providing for any category the most relevant approaches proposed in the literature. Finally, advantages and drawback are pointed out in order to evaluate the real impact of introducing the self-adaptive in the common wireless sensors networks, showing a substantially gain in terms of performance for a very dynamic system while an inefficiency when the systems do not face with recurring changes.

References

[1] N. Marchetti *et al.*, "Self-organizing networks: State-of-the-art, challenges and perspectives," in *Proc. of COMM*, 2010.

[2] C. Sengul *et al.*, "A survey of adaptive services to cope with dynamics in wireless self-organizing networks," *ACM Comput. Surv.*, 2012.

[3] L. Subramanian and R. Katz, "An architecture for building self-configurable systems," in *Proc. of MobiHOC*, 2000.

[4] B. Pasztor *et al.*, "Selective reprogramming of mobile sensor networks through social community detection," in *Proc. of EWSN*, 2010.

[5] A. Cerpa and D. Estrin, "Ascent: adaptive self-configuring sensor networks topologies," *Mobile Computing, IEEE Transactions on*, 2004.

[6] S. Sehic *et al.*, "Copal-ml: A macro language for rapid development of context-aware applications in wireless sensor networks," in *Proc. of SESENA*, 2011.

[7] A. Taherkordi *et al.*, "Supporting lightweight adaptations in context-aware wireless sensor networks," *Proc. of CAMS*, 2009.

[8] M. Steine *et al.*, "Proactive reconfiguration of wireless sensor networks," *Proc. of MSWiM*, 2011.

[9] M. Zimmerling *et al.*, "ptunes: Runtime parameter adaptation for low-power mac protocols," in *Proc. of IPSN*, 2012.

[10] S. Rizvi *et al.*, "Use of self-adaptive methodology in wireless sensor networks for reducing energy consumption," in *Proc. of ICIET*, 2007.

Mikhail Afanasov
Sonda Bousnina
Giuseppe Massari
Vincenzo Sciancalepore

Politecnico di Milano
Italy

Anonymous Reviewers
Guest Editor Team
SASS'13

Dear reviewers, dear editors,
We would like to thank you for your feedback on our submission, as well as for the insightful comments, which allowed us to further improve the quality of our paper.
We have done our best to address all the raised concerns.

Review 1

Q: "I would suggest to reduce a bit the "State of the art" section (which takes about the 70% of the paper length) and, if possible, introduce a section with the description of at least one application among those described in the papers considered for the survey".
A: We disagree on this point, since state of the art is the core of a survey. Regarding the application, since the approaches are application-specific it is very hard to provide a single example that would cover all the problems from all the perspectives.

Q: "In this context, … to an unpredicted environmental dynamics": the meaning is understood but the sentence should be rephrased to improve clarity.
A: This sentence is rephrased in order to improve the clarity for the reader

Q: "Focusing on the one of the directions … use them all together": can Architecture and System directions be considered somehow linked? See also my previous comment.
A: As we shown in the paper, architecture level deals with topology dynamics, while system level cope with internal parameter of the node in WSN. These two levels are orthogonal, so we can safely combine them in one effective solution.

Q: "There also could be an intermediate state, when the old behaviour is not applicable in the new situation, but the new behaviour is not chosen yet.": could you please clarify what it is supposed to happen at system level in such particular condition? Are there strategies to avoid this?
A: The system simply misbehaves, and cannot be relied on. This is a critical problem, but to treat it is out of the scope the report.
However, the sentence has been rephrased as follows:
"*There also could*
be a transitional state, where in the environmental conditions are changed and
the old behaviour is no more suitable, but the new behaviour is not calculated
or applied yet."

All the typos reported have been fixed, and sentences have suitably rephrased.

Review 2

Q: There must be improvements of English. For instance "the sensors' coverage". The "s'" structure is never used with anything inanimate.
A: Fixed by rephrasing the sentence.

Q: In the introduction, there is no description of current technologies (nonselfadaptation ones) that are used in WSN. So it is unclear how this problem is solved today, with what methods.
A: Non-adaptive methods have been not taken account since they are off topic for this report.

Q: There are use-cases for each self-adaptation layer, but there is no example that combines those layers together. However, in the evaluation the authors claim that the directions are independent, so maybe there is no such an example.
A: The ASCENT approach is an example. The report has been corrected to highlight this point.

Review 3

Q: Section III (Evaluation) of the manuscript need to be revised and rephrased. Although, authors tried to provide evaluation of existing work, but still few unclear concepts. Needed a major revision in section III.
Need further elaboration on each evaluation remarks provided in manuscript
A: Section III has been completely reviewed accordingly.

Minor issues reported have been also fixed.

Review 4

Q: Unclear distinction in three research directions
A: Adaptivity on system level focuses on the internal state of the node such as reconfiguration and resource management. Architecture level adaptivity is considered and implemented on architectural level, regardless the platform and application it will be used for. Application level adaptivity does not care about network topology or routing protocol, but performs on top of it. It is discussed in the paper in details. Anyway, we add in the report a statement saying that it is quite common that an approach covers more than one level/perspective.