Node-Level Optimization of Wireless Sensor Networks

Simone Campanoni, William Fornaciari
Politecnico di Milano, Dipartimento di Elettronica e Informazione
P.zza L. Da Vinci, 32
20133 Milano, Italy
campanoni@elet.polimi.it, fornacia@elet.polimi.it

Abstract— **A methodology for WSN planning should produce a suitable placement of sensors to simplify the acquisition of the data relevant for the application. The goal of this paper is to present a strategy to formally specify the system level characteristics of the events to be monitored and to identify a proper set of sensor-position pairs, tailored to provide the required sensing capabilities. In particular the focus is on the node level optimization, where the designer has to face with the problem of clustering sensors onto the same board based upon a cost-performance tradeoff.**

I. Introduction

Building a WSN become apparently as simple as composing COTS (Component Off The Shelf) [1] [2]. Since a few years, in literature appeared a number of proposals regarding simulation and deployment of WSNs. However, from the best of our knowledge, none is addressing with a proper and formal extent the capability of the network to capture the events to be monitored. Their focus is frequently related to the optimization of the cost or to verify other properties like networking exploitation, power consumption, robustness of the connection layer or the analysis of the models of computation (middleware).

As far as the *offline planning* is concerned, which is the case this paper is focusing on, the proposals usually deal with only one single objective (e.g., coverage) or in some cases with lifetime in terms of energy consumption. The sensing model is normally built around flat squares, and only few proposals cope with simple obstacles [3] [4]. Noticeable solutions regards data-centric high level representations of the network behavior. TinyDB [5] has been a pioneer effort enabling a SQL-based interface to the sensed data. GSN [6] is another proposal based on XML and SQL, taking into account the problem of dynamic reconfiguration of the system. A declarative approach to the network description has been considered in [7], where a dialect of Datalog is used for both data acquisition and transmission management. On top of such internal data representations, an engine to recognize events can be implemented. In [8] Symblic Aggregate approXimation (SAX) is used as an algorithm for detecting complex events by analyzing the patterns related to the sensed basic parameters.

The scope of the work here presented is a wide class of applications where, in addition to the typical monitoring capabilities, it is also required a prompt highlight of the occurrence of particular events. Our methodology requires to: *specify* the characteristics of the events of interest; *select* a proper set of sensors tailored to catch such events and to *insert* the sensors in the environment ensuring the capturing of the desired events while optimize design goals/constraints. Such design stages are part of the SWORDFISH design framework [9], whose node level optimization strategy is here presented.

The paper is organized as follows. Section II outlines the architecture of SWORDFISH. Section III discusses the model of the events to be recognized and the design flow to create the reference WSN. Sections IV, V and VI are more related to the physical constraints/optimizations of actual implementations. Concluding remarks are drawn in Section VII.

II. The WSN Design Environment

SWORDFISH supports system-level design of WSN applications, by addressing the following problems.

- *Verification.* The goal is to determine the occurrence of a set of events (e.g., fire in a defined region, temperature and humidity over a certain threshold for a time window, etc.) by exploiting the *capabilities* of a given sensor network.
- *Sensitivity Analysis.* Evaluation of the impact of some variation of sensors, environment and network properties, onto the performance of a WSN. Examples are fault tolerance w.r.t. sensors and network errors, changing of sensor location, influence of the observation time, etc.
- *Design/Planning.* The task is to discover the optimal sensor network capable to identify the events while maximizing a goal function and fulfilling possible constraints.

The SWORDFISH architecture includes a set of modules outlined in the following, allowing the users to describe actors and design goals (see Tab.I) and to support system verification.

TABLE I. Main Editors of SWORDFISH.

Editor	Purpose
Environment	It allows defining a model of the environment with the possibility to specify constraints such as position and type of some sensors.
Sensor	It is the mean to obtain the analytic representation of the sensing nodes. The model of the node includes also information such as cost, type of sensors, energy consumption, accuracy, speed, …
Network	It provides a model of the available connection channels among nodes.
Predicate	It allows the user to specify via logic formulas the properties to be verified when a given event occurs.
Event	It supports the description of the events in terms of variation of some physical parameters to be sensed, along with their timing characteristics.
Optimization	It allows the designer to specify and tune the goal functions and the formal model of the network properties/constraints.

The *Simulation Kernel* is the engine which, based on a simulation of the event occurring, modifies the configuration of the world model accordingly. This allows feeding the sensor node models with the real (location aware) data of the world, including their dynamics. Both the physical parameters of the environment and the events to be monitored can be jointly modeled and verified by a module called *Predicate Analyzer*. The last module is named *Planner*. It is the main module for both verification and network design. It allows to formally verifying that a given WSN is able to capture a set of events as well as to support the building and optimization of the overall network according to the selected policies and goals.

Based on the application requirements, the first steps for the user are formally defining the events to be captured and possibly some optimization goals/constraints. Network properties and sensor behavior can be also specified, in the case of default settings are not considered suitable. According to the existing model of the environment, the events are then "fired" to get a profiling of the evolution of the physical parameters corresponding to the events. Such results are then used as a testbench to compare the performance of alternative WSNs in terms of sensing capabilities. Useful information for optimization can be gathered by analyzing the sensitivity of the network over the variation of parameters like clustering of sensors, as shown in the following Sections.

III. Preliminary Steps of the Design

The model of the environment is 3-D where (x,y,z) coordinates belongs to a user-defined grid. Before starting the exploration of the WSN design space, there are three preliminary steps to be carried out: i) definition of the *purpose* of the network; ii) identification of the *benchmark*; iii) modeling of the *hardness* to recognize physical parameters corresponding to an event.

The first activity is the definition of an overall *Sensing Goal* (SG) for the WSN, that is a multi-value logic formula composed of some predicates Pr, each corresponding to an event. For example Water(x,y,z,magn,trend) is a plug-in modeling the presence of *water* in the point (*x,y,z*), starting from a given *magnitude* and with a specified *trend* over the time. A predicate Pr is an instance of Water applied to a specific point. An example of SG is (1).

$$\text{SG=Temp}(0,1,2,20,\text{const})\ \text{AND Humidity}\ (3,4,5,70,\text{const}) \quad (1)$$

Such SG means that the WSN has the goal to discover the concurrent presence of the events of having a certain Temperature (20) and Humidity (70) in two points (0,1,2), (3,4,5) of the environment.

The second step is the characterization of the changing in the environment whenever the events occur, namely the identification of a testbench to evaluate the WSN performance. To this purpose, based on the (user defined) *fp* sampling rate of the environment simulator, a profiling stage is triggered by firing each of the defined events, namely running the Pr-related software plugins. At the end, ∀(x,y,z), and ∀ Pr of SG, all the data patterns are obtained.

There are at least other two problems the designer has to face with during WSN design: *selection* of the type of sensor and sensors *placement*. In fact, the target is to discover a positioning of the sensors, maximizing the capability of the WSN to recognize the events, i.e. maximizing the SG. The former question impacts mainly on the feasibility of designing a WSN capable to recognize the events encompassed by the SG. The latter is related to the dissemination of sensors in order to enhance their possibility to satisfy the Pr composing the SG, i.e. improving the performance of the system.

Our first concern in the design flow is ensuring that it exists a solution to the SG, using a proper set of sensors that is incrementally built up and significantly optimized by sharing sensors among the set of Pr (specified in the SG) to be verified. Then, this set of candidate sensors are placed in the environment taking into account the information coming from a configurable *hardness* function. In such a way it is guaranteed to obtain a WSN satisfying the SG with a quasi-optimal cost, with runtimes in the order of a few seconds.

As far the positioning of the sensors is concerned, we defined a *hardness* function Hard(x,y,z,Pr) modeling the difficulty in evaluating Pr in a given point (x,y,z).

$$\text{Hard}(x,y,z,\text{Pr})= \text{Hs}(\text{PPr},t)/\ C\{(\text{PPr},t),\ \text{Pr}\} \qquad (2)$$

Calling PPr the *profiling output* of Pr, i.e. the data pattern associated to Pr obtained during the initial profiling, Hs(PPr,t) depends on the type of sensor (corresponding model) and relates to the difficulty to recognize the event Pr within the time frame of a profiler sampling rate (1/fp). For example for a slow temperature sensor can be hard (or even impossible) recognizing T-ramps moving faster than its cutting frequency. The other term, C{(PPr(x,y,z), Pr}, is the confidence to infer the truth of Pr based on the sequence of the physical variations defined via PPr.

Of course, any positioning strategy for the sensors attempts to place the sensor where *Hard* is low, i.e. where it is easier and reliable recognizing the Pr composing SG. More formally, it is selected the Si to be assigned to the predicated Pj, such that $\lfloor dSG/dPj \rfloor$ ∀ Si available, is minimum.

The implemented algorithm actually starts considering only the models of the available types of sensors and the predicate Pr to be satisfied, with possibly additional constraints (e.g., cost figures) that can be provided by the users within the sensor plugins. Then, the minimum set of sensors capable to recognize physical parameters to satisfy all the Pr is discovered and initially allocated to the most relevant predicates (in the SG sense). Based on this initial allocation, that is a pre-condition to satisfy the SG, the analysis of possible sensor sharing takes place. The end of the process produces a solution employing the minimum set of sensors covering all the predicates, using a quick heuristic producing a configuration that in most of the cases it is also the optimum.

To represent how a given sensor is actually capable to capture its target events from a position (xp,yp,zp), a proper metric (3) has been defined, called *confidence*.

$$\mathrm{Conf} = 1-(\mathrm{Hard}(xp,yp,zp,Pr))/\max \mathrm{Hard}(x,y,z,Pr)) \qquad (3)$$

Where Pr is the predicate corresponding to the event, Hard (xp,yp,zp,Pr) is the hardness calculated in the candidate point for the sensor positioning and max Hard(x,y,z,Pr) is the maximum hardness within the considered environment. Note that values of confidence closer to one means that the position of the sensor is approaching the best existing in the environment to satisfy Pr, while lower values corresponds to critical points; this latter case can trigger the search for a better positioning or the increasing of the sensor set cardinality. More details on the sensor level design activities can be found in [9].

IV. Modeling of Nodes

The output of the SWORDFISH-based feasibility analysis is a set of {sensor, position} pairs tailored to optimize cost-effectiveness and performance (sensing goal) of the WSN. On the other hand, realistic design and deployment typically requires simplifying both the hardware and the architecture of the network, while maintaining effectiveness and performance.

To cope with such problem, in the following section our PESCA (Pareto Efficient Solution Clustering Algorithm) approach is outlined. It produces a quasi-optimal clustering of the sensors by identifying a proper mapping of the sensor set onto boards, like those available on the market [2].

In general, each board hosting sensors includes the following sections: PCB/package; power supply and energy management; radio (RX/TX); control/processing Unit (CU); connectors/Interfaces; one or more sensors. Based on our experience in realizing PCB-level embedded systems and on market availability of sensing modules, we found reasonable adopting the model (9) for the cost of each board (*node*).

$$NodeCost = Const + K \times \log(N) + \sum_{j=1}^{SensorTypes} (SC_j \times NumS_j) \qquad (9)$$

Where, for each board, N is the overall number of sensors, $NumSj$ is the number of sensors of a given type j, $SensorTypes$ is the number of possible types of sensor, and SCj is a cost of a sensor of type j.

In other words, there exist a variable cost which is related to the type and number of sensors in a linear manner and a processing cost that is logarithmic, due to the typical price trends of CPUs and microcontrollers. Furthermore, the cost of PCB and packaging is less than linear against the number of sensors (close to a constant), while radio and power supply is fairly stable over a wide range of possible sensors cardinality.

To increase the generality over different suppliers, we partitioned the available sensors into classes, to capture their relative cost, instead of considering the absolute values. Concerning the cost of the network, we assume a constant value for wireless connections (typically built-in in commercial nodes). Actually, some influence of the network topology should be considered in the case of some gateway nodes, mastering hierarchies of sensors patches, were identified. In such a case there is an additional cost related to the wired connection or the use of other long-range radio communication standards/modules (e.g., GPRS).

V. Board-Level Design

The clustering of the set of sensors identified by the SWORDFISH framework is a multi-stage process, including the following topmost activities: *compatibility* analysis between all the possible pairs of sensors; identification of the *boundaries* of the clustering problem (worst and best case); *generation* and *evaluation* of the candidate solutions.

A. Compatibility Graph

At the beginning, the user (e.g., by accepting default settings) is required to specify constraints on the possible clustering of different sensors onto the same board. Based on these information, an Interference Graph G=<N,E> is built, where nodes *n* are sensors and an edge *e* among nodes represents a possible sensor interference to be avoided.

Note that the interference is not only related to the nature of the sensors, as specified by the user. In fact, two sensors can have no shared position where both sensors are still able to discover the associated set of events. This latter case can be discovered by using the Hard function; in fact, the Hardness of a point to discover two events *e1*, *e2* is the sum of the Hardness of both sensors computed in that point.

Once the interference graph is completed, it is possible to identify the compatibility graph G', which is its complementary graph G'=<N,E'>, gathering all the feasible solutions. Note that any possible clustering of sensors, cannot but be a clique of the compatibility graph. In fact, all the sensors hosted by the same board must be cross-compatible.

The next step is the computation of all the maximal cliques of the compatibility graph G'. Since this type of activity is recognized to be a NP-hard problem [10] we adopted some heuristics to speedup the computation.

B. Coverage of the Sensor Graph

To better understand the optimality of placing a certain group of sensors onto a board, we focused at the beginning on the boundaries solutions, considering the cardinality of the cliques (not their cost). At the lowest level, each sensor corresponds to a board. As far the best case is concerned, the biggest clique has been computed, and then the same action has been performed on the remaining graph, and so on. At the end we obtain a partitioning of the compatibility graph, where its cliques are those clustering the maximum number of compatible nodes. The design space spanning between the two boundaries cases so identified, contains a number of possible solutions that is exponential with the sensor cardinality. We attacked the analysis of a so wide solution space through an heuristic structured into a couple of consecutive steps.

1. Starting from the best case above described, solutions are generated by creating a new list of cliques where one sensor has been extracted from the biggest clique, to create a new single-sensor board. The same activity is then repeated, considering the biggest cliques at any iteration. At the end of such a process, a wide set of possible solutions is obtained, ordered for relevance i.e., in terms of cardinality of the biggest clique.

2. All the possible pairs of the above solutions are taken into account for possible merging, while verifying the compatibility of the new sensor set.

At the end of the second steps, the recombining of cliques not maximal in terms of cardinality, allows the optimizer to consider solutions more *orthogonal*, possibly characterized by a lower board-level cost.

C. Selection of Solutions

This step takes into account the candidate WSNs under the Pareto standpoint. The task of the PESCA algorithm is to find out a solution to the multi-objective clustering problem, considering two metrics: the *cost* of the solution and the functional quality, i.e. its *performance*.

The cost of a solution (set of boards) is evaluated through the cost model (9) described in Section IV, that is depending on the number and type of sensors associated with each partition. Concerning the performance, the badness of a solution is computed by exploiting the Hardness functions *Hij(x,y,z)* of the event *i* covered by the sensor *j* belonging to the same board. Thus, the hardness of the entire WSN is:

$$H(x,y,z) = \sum_{j \in WSN} \sum_{i \in J} H_{ij}(x,y,z) \qquad (10)$$

The badness of the WSN is evaluated, and its minimum corresponds to a point where the positioning of the board is optimal. This new location, which is shared by all the sensors on the same node, is the best to ensure that all the events associated with the sensors can still be captured after clustering. Note that the solution so discovered is a Pareto efficient solution. In fact, all the solutions "dominated" within the *<cost, performance>* optimization space of the WSN are discarded during the population of the design space.

VI. Experimental Results

The PESCA approach to the clustering of sensors has been evaluated considering some validation scenarios extracted from the multi-partner projects supporting this work [11] [12].

In Fig.1 it is reported the analysis carried out on a complex WSN: SG of 16 basic predicates producing an optimal network with 18 ideal sensors. The plot reports the Pareto solutions and figures out the influence of the fixed costs (linked to the volume/standardization of boards) against the overall cost and performance (1/Hardness) of the clustered WSN, varying the fixed cost of the board.

It is possible to observe the impact on performance and cost associated with the spreading vs clustering of sensors: the proposed quantitative analysis produces a significant value added for the designer when the tradeoff is not so *evident*. In such a way, the driver may be not only the cost, but also the capability of the WSN to fulfill the initial application requirements.

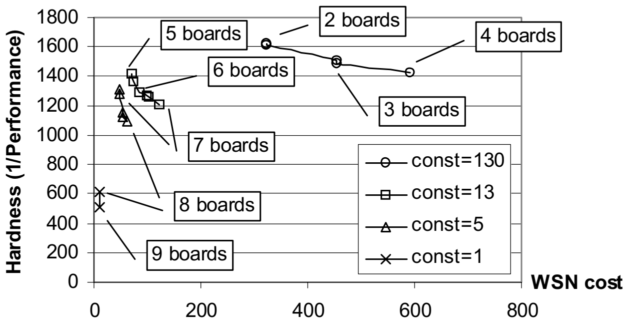

Figure 1. Pareto frontier of the cost-performance tradeoff.

VII. Conclusions

The paper presented the overall methodology of SWORDFISH to design WSN at system-level. Particular attention has been paid to the clustering of sensors onto a reduce set of boards with realistic architectures. The obtained results are promising, and some of the verification and top-level analysis and design capabilities have been addressed by considering also representative use-cases [11] [12]. It has been shown how it is possible to optimize the WSN while ensuring that the original user' goal has been fulfilled. The examples reveal that many side-effects of changing the behavior of the WSNs and sensor positioning (or clustering) produces significant changing in the sensing goal that are hard to be managed by a human designer, without the support of a methodology and a tool like SWORDFISH.

References

[1] Akyildiz I.F.; Weilian Su; Sankarasubramaniam Y.; Cayirci E. 2002. A survey on sensor networks,IEEE Comm. Mag., vol. 40, n. 8, pp. 102-114, Aug 2002.

[2] www.moteiv.com, www.xbow.com

[3] S.Dhillon, K.Chakrabarty, S.Iyngar. *Sensor placement for grid coverage under imprecise detections.* Proc. Of Int. Conf. on Information Fusion, July 2002, pp 1581-1587.

[4] A.Howard, M.Mataric, G.Sukhatme, *An incremental self-deployment algorithm for mobile sensor networks.* Autonomous Robots-Special Issue on Intelligent Embedded Systems, Vol.13(2), 2002. pp 113-126.

[5] S.Madden, M.Franklin, J.Hellerstain, W.Hong, TinyDB: an acquisitional query processing system for sensor networks, ACM Trans. on Database Sys, vol.30 (1), 2005.pp122-173.

[6] K.Aberer, M.Hauswirth, A.Salehi, *A middleware for fast and flexible sensor deployment,* Proc of 32nd Int. Conf. on VLDB, 2005. pp 1199-1202.

[7] D. Chu, L.Popa, A.Tavakoli, J.Hellerstein, P.Levis, S.Shenker, *The design and implementation of a declarative sensor network system*, Proc. of SenSys'07, November, 2007. Sydney, Australia.

[8] M. Zoumboulakis, G.Roussos, Escalation: Complex Event Detection in Wireless Sensor Networks, LNCS Smart Sensing and Context, Vol. 4793/2007, October 2007.

[9] S.Campanoni, W.Fornaciari, SWORDFISH: a Framework to Formally Design WSNs Capturing Events, In Proc. of IEEE SoftCOM'07, Split-Dubrovnik, Croatia, September, 2007.

[10] M. R. Garey, D. S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness (Series of Books in the Mathematical Sciences), W. H. Freeman Publisher, 1979. ISBN 0716710455.

[11] Adaptive Infrastructure for Decentralized Organizations - ARTDECO, Min. for the National Research, 2006-2009. http://artdeco.elet.polimi.it

[12] WASP (Wirelessly Accessible Sensor Populations), EC-Funded IST project, http://www.wasp-project.org/