Multi-Level Design and Optimization of Wireless Sensor Networks

Simone Campanoni, William Fornaciari
Dipartimento di Elettronica e Informazione
Politecnico di Milano
Via Ponzio 34/5, 20133 Milano, Italy
{campanoni, fornacia}@elet.polimi.it

***Abstract*—This paper proposes a methodology to off-line planning of WSNs (Wireless Sensor Networks) by addressing the problem in a multi-layer manner. At the *sensor* level a model is described to properly select and distribute the sensors in the environment. To optimize the cost and deployment of realistic WSNs, further design activities are proposed at an intermediate level, targeting *board*-level clustering of sensors. Finally, it is presented a methodology to hierarchically organize the set of sensors in patches with an additional gateway-level communication layer, to take into account also possible scaling of the application complexity. Particular emphasis is put on the cost modeling and on ensuring the correct behavior of the WSN against variation of parameters like sensor position, protocol, cost, etc.**

Keywords: Wireless Sensor Networks, System-Level Design, Cost-Performance Tradeoffs, Cost-Model, Design Frameworks.

I. Introduction

A design flow spanning from the abstract specification of the WSN down to the node/network architecture, requires considering a number of multi-level optimization strategies, whose individual effectiveness can vanish if all the various aspects (e.g., communication protocol, programming model) are not considered concurrently, within a comprehensive system-level design methodology [1]. The increasing size and heterogeneity of the applications, in recent years exacerbated the WSNs design problem and spurred the interest of the research community in network planning. The focus is typically on optimizing the location of the sensors to maximize their collective coverage of a given region. This challenge has been tackled using several approaches, such as integer programming [2] or greedy heuristics [3][4][5] to incrementally deploy the sensors. In addition, methods used for robotics have been adapted to the purpose of deployment [6]. Some authors paid particular attention to the connectivity problem from a more formal standpoint [7]. There exist adaptive techniques considering scenarios where multiple sensors are needed, accounting for a possible real-time deployment [5][3].

As far *offline* planning is concerned, which is the focus of this paper, the proposals usually deal with only one single objective (e.g., coverage) or in some cases with lifetime in terms of power consumption. The sensing model is normally built around flat squares, and only few proposals cope with simple obstacles [3] [4]. From a more abstract standpoint, the problem of designing WSNs produced noticeable solutions identifying data-centric high level representations of the overall network behavior. For example TinyDB [8] has been a pioneer effort enabling a SQL-based interface to the sensed data, while considering the need of achieving a power efficient processing and routing of query data. GSN [8] is another proposal based on XML and SQL as data specification and data manipulation languages, taking into account the problem of dynamic reconfiguration of the system. A declarative approach to the network description has been considered in [10], 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 [11] Symblic Aggregate approXimation (SAX) is used as an algorithm for detecting complex events by analyzing the patterns related to the sensed basic parameters.

From a pragmatic standpoint, the availability of a design framework is of paramount importance to face the complexity of the design space. We counted about 40 proposals in literature, where the most mature projects, having also a web site for download of articles and tools, are (references are omitted for conciseness): Ns-2, SensorSim, EMStar, OPNET, OMNET++, Avrora, TOSSIM, VisualSense/Ptolemy, SENSE, J-Sim. These environments allow defining a network of sensing nodes along with the specification of a networking layer tailored to consider the peculiarity of wireless-oriented protocols. All of these environments provide hooks to add functionalities linking ad-hoc code. The majority of them provide some form of discrete event simulation and Ptolemy also allows mixing different domains of computation and aspects like mobility of nodes. However, to 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 by simulation other properties like power consumption, robustness and congestion of the connection layer or the analysis of the models of computation (middleware).

The focus of our investigation is a wide class of applications where, in addition to the typical monitoring capability, a prompt highlight of the occurrence of particular events is required. The requirements analysis of such scenarios is carried out within the context of two cooperative research projects (ARTDECO [12] and WASP [13]). The planning of WSNs for such applications is a off-line activity and requires to: i) specify the characteristics of the events to be discovered; ii) select a proper set and type of sensors to enable the capturing of such events; iii) embed the sensors in the environment in a way to ensure the capturing of the desired events while optimizing some design goals.

Our target is first of all to make sure a priori the existence of a feasible solution to the sensing problem, with the accuracy required by the application. Then, by exploiting the capabilities of the SWORDFISH (Sensor netWORks Development Framework Integrating Simulation and Hardware optimization [14]) optimization engine, the WSN is refined according to design constraints and user's goals. In particular, the target of this paper is to show how it is possible to design a WSN taking into account realistic constraints, such as the necessity to simplify the realization and deployment via a proper aggregation of sensors onto a set of boards, while ensuring acceptable performance degradation.

The paper is organized as follows. The next two sections summarize the design flow implemented in SWORDFISH, to show the steps necessary to define the optimal choice and localization of sensors. The rest of the paper is devoted to show new results concerning the strategy to cluster the set of sensors identified as optimal by SWORDFISH: Section IV presents the cost model for the network nodes, Section V the optimal clustering of sensor at board-level and Section VI discusses and compares four strategies to determine the positioning of gateways, in the case of the size of the network requires a hierarchical organization. Experimental results are reported in Section VII, while some conclusions are drawn in the last Section.

II. Multi-Level Design Methodology

One of the objectives of the proposed design methodology is to create a design flow for WSN offline planning, which is scalable with the application complexity. To this purpose, the first step is to identify a proper set of sensor-position pairs, considered optimal to achieve the desired behavior of the WSN (see Tab.I).

TABLE I. Design Space Exploration

Level	Activities	Main Objectives
Sensor	Selection and positioning of sensors set; sensitivity analysis	Optimize # sensors Satisfy sensing goal
Board	Aggregation of sensors onto some boards; sensitivity analysis	Optimize realization cost and deployment
Subnet	Identification and positioning of gateways; protocol selection	Optimize overall cost and performance

This initial solution is the baseline for any architectural design space exploration. The next optimization is related to the aggregation of the previously identified sensors set onto boards, while meeting effectiveness and acceptable degradation of the WSN behavior. The outermost layer accounts for complex sensor networks, where some of the nodes (boards) have to manage hierarchies of sensors/subnets.

The focus of this paper is on *board* and *subnet* level design space exploration. Sensor-level design is only sketched to provide a complete perspective of the methodology (see [14] for details). The support to such system-level design seats on a modular framework, called SWORDFISH, whose structure is depicted in Fig.1. It is composed of a set of modules allowing the users to describe the actors (environment, sensors, network and events) and the design goals (desired behavior of the WSN and optimization parameters).

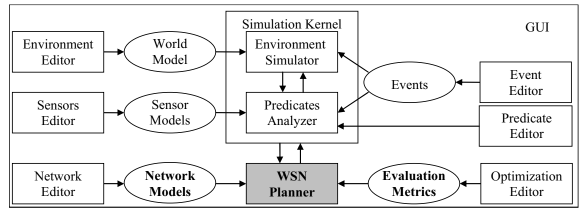

Figure 1. The SWORDFISH Toolsuite.

SWORDFISH is conceived to support the users during WSN planning, by addressing the following problems [14].

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

Based on the application requirements, the first step for the user is formally defining the events to be captured (see Section III) and possibly some optimization goals/constraints (Predicate, Event and Optimization editors). Network properties and sensor behavior can be also specified (Network and Sensor editors), if the default settings are not considered suitable. According to the model of the environment (specified using the Environment editor), 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 sensing capabilities of alternative WSNs. The Predicate analyzer and the selected optimization goals are extensively used by the Network Planner to explore the design space. Useful information for optimization can be gathered by analyzing the sensitivity of the network over the variation of parameters like observation time [14] or sensor accuracy. In this paper we will describe only the network planner module (gray box in Fig.1), whose goal is to cluster sensors and identify the gateways nodes, based on a previous definition of the optimal sensor set dissemination.

III. Sensor-level design

The methodology we propose to identify a proper dissemination of the sensors within the environment specifies the purpose (behavior) of the network, analyzes the benchmark against which the WSN have to be checked and, finally, quantifies the difficulty of sensing the physical parameters characterizing the events to be recognized.

To model the complex event or condition the WSN is supposed to recognize [14], we refer to a Sensing Goal (SG), which is a multi-valued logic formula, where each basic event is represented by using a sub-predicate Pr (implemented via software plug ins) having the general structure Event(x,y,z, magnitude, trend). For example, (1) means that the goal of the WSN is to recognize the concurrent presence of Water in two points (1,2,0) and (3,3,5) of the environment, with a constant magnitude of 20 and 30, respectively. In practice, the overall SG of a WSN can includes tenths of basic events/predicates.

SG=Water(1,2,0,20,const) AND Water (3,3,5,30, const) (1)

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

The other two problems the designer has to face with during WSN design are: *choice* of the type of sensor, and optimal sensors *placement*. As described in [14], SWORDFISH produces a selection and the optimal placement of sensors to enhance their possibility to satisfy the Pr composing the SG, i.e. improving the performance of the WSN. To drive such process, we defined a hardness function Hard(x,y,z,Pr) modeling the difficulty to evaluate Pr in a given point (x,y,z). Calling PPr the "profiling output" of Pr, i.e. the data pattern associated to Pr obtained during the initial profiling, we define the Hard function (2).

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

Where 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 the profiler sampling rate (1/fp), and 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. 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, that we named confidence [14]:

Conf(event)=1-(Hard(xp,yp,zp,Pr))/max Hard(x,y,z,Pr)) (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. On the contrary, lower values correspond to critical points; this latter case can trigger the search for a better positioning or the increasing of the sensor set cardinality. In the case a sensor is shared between a set of events corresponding to a group of predicates, the confidence is calculated by adding the hardness of all the predicates Pr the sensor have to cover.

IV. COST MODEL OF THE NODE

The sensor-level produces a set of {sensor, position} pairs tailored to optimize cost-effectiveness and capability to fulfill the 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, by exploiting boards hosting multiple sensors. This constraint necessarily modifies the optimal positioning of sensors, with the risk of side effects on the desired WSN behavior.

To cope with such problems, the PESCA (Pareto Efficient Solution Clustering Algorithm) approach is here outlined. The goal is to produce a suitable clustering of the sensors by identifying an optimal mapping of the sensor set onto boards, like those available on the market [15][16] or emerging from research projects [12][13]. In general, each board hosting sensors includes at least the following sections: PCB/package; power supply and energy management; radio (RX/TX); control/processing Unit (CU); Connectors/Interfaces; one or more sensors. Furthermore, the cost of radio interface can vary, especially in the case the WSN architecture/complexity requires the presence of some gateways managing patches of sensors. Depending on the application, three different cases can be envisioned:

- New ad-hoc boards are realized for the application.
- Use of off-the-shelf boards already existing on the market.
- Customization of boards, e.g. by adding daughter boards to create gateways or to add specific sensors.

Based on our experience in realizing PCB-level embedded systems, on market availability of sensing modules and the results emerging from the application scenarios defined in [12][13], we founded reasonable to adopt a general model of monetary cost for each board (node). We observed that 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 CPU 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. In summary, the cost of a node is the sum of the following terms:

- NRE/n. It is the quote of the initial design (Non Recurrent Engineering) cost spawned over n copies of the same board.
- Const (1+(markup/n)). Accounts for the discount achievable when sourcing components with significant volumes.
- K*log(N), takes into account the processing power; N is the number of sensors of the considered board.

- C_{MAC} is the cost of implementing the level 2 of the ISO/OSI stack.
- $\sum_{j=1}^{SensorTypes}(SC_j \times NumS_j)$ is the cost of the sensing. 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.
- C_{GTW} is a fixed cost in the case the node acts as a gateway.

To consider different suppliers, we partitioned the available sensors into classes, to capture their relative cost, instead of considering the absolute values (see Tab.II). Concerning the cost of the network, we assume a constant value depending on the protocol (C_{MAC}) for wireless connections (typically built-in in commercial nodes). Furthermore, some influence of the network topology should be considered in the case of some gateway nodes, managing 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 and modules (C_{GTW}). In this paper only wireless connections have been considered for the gateways, whose additional cost is not strongly dependent on their positioning, but mainly on the adopted standard

TABLE II. RELATIVE COST OF SENSORS (EXAMPLES).

Class	Type of Sensor	Relative Cost
1	Temp, Humidity, Light,…	1
2	Pressure, Presence, Resistance, Temperature (high accuracy), …	2
3	Gyroscope, Accelerometer …	3
4	Pollution (gas, water), complex MEMS,...	5
5	High-End	10

V. BOARD-LEVEL DESIGN

The clustering of the set of sensors identified by SWORDFISH 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 selection of the candidate solutions.

A. Compatibility

At the beginning, the user (e.g., by accepting default settings) is required to specify constraints (taboos) 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* between two nodes represents a possible sensor interference to be avoided. Note that the interference is not only depending on the nature of the sensors, as specified by the user via taboos. In fact, two sensors can have no shared position where both sensors are still capable 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 compiled, it is possible to identify the compatibility graph G', which is its complementary graph G'=<N,E'>, gathering all feasible solutions. It worth noting, 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 compatible with each other. 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 [17] we adopted some heuristics.

B. Coverage

To better understand the optimality of placing a certain group of sensors onto a board, at the beginning we focused on the boundary 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 a heuristic structured into a couple of steps.

- Starting from the best case above described, new 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 generated, ordered for relevance i.e., in terms of cardinality of the biggest clique.
- All the possible pairs of the above solutions are taken into account for possible merging, while verifying the compatibility (taboos) 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 less homogeneous, possibly characterized by a lower board-level cost

C. Comparison 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 and the functional quality, namely its performance.

The cost of a solution (set of boards) is evaluated through the cost model described in Section IV, which is depending on the number and type of sensors associated with each partition. Concerning the performance, the quality 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 (4):

$$H(x,y,z) = \sum_{j \in WSN} \sum_{i \in J} H_{ij}(x,y,z) \qquad (4)$$

The hardness 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. Some quantitative examples are discussed in Section VII.

D. Use of existing boards

The objective of mapping an ideal sensor network (where each sensor is a node) onto a realistic set of boards, can have different flavors. In the above section we argued how it is possible to derive a clustering in presence of some degree of customizability at board level. Our toolset implementation, even if it is not detailed in the paper due to space limitation, solves also the problem of mapping the sensors onto a catalog of standard boards. In such case the cost model is simplified and the strategy we follow resembles the solution to the problem of Instruction Selection of the compilers [18].

VI. SUBNET-LEVEL DESIGN

For a number of the considered use cases, the coverage of the area cannot be performed through a single starry patch of sensors. Roughly speaking, considering an average of 3-7 sensors per board, it is reasonable that for every group of at most 6-12 boards, one of them has to become also the gateway of a hierarchy of nodes (boards). In particular, for one of the scenarios [12], it is required to disseminate possibly over a hundred of boards. The problem afforded in this section is then twofold: computation of an optimal *partitioning* of the network and *selection* of the nodes which will operate also as *gateways* for the sets of boards composing the partition. Our strategy produces the optimal solution in the case of small-medium size WSN, while for large-size applications we propose heuristics producing near-optimal network architectures, with runtime still in the order of tenths of minutes.

The *input* of the problem is the set of boards along with their positions, as described in Section V. The *output* is a set of Pareto-efficient solutions, specifying: i) the board selected to become also gateways; ii) the partitioning of the boards set in subnets assigned to a single gateway; iii) the overall throughput of the WSN, measured as the mean or the minimum of that pertaining the identified subnets; iv) the new cost of the WSN. Note that each subnet is described not only by its set of nodes, but also by the level-2 protocol (e.g., 802.11 rts/cts, 802.15.4, etc). To this latter purpose, we analyzed (Section VII) the worst case (saturation), using well assessed literature models and assumptions [19][20].

A. Gateway selection

The target is to choose the boards to be promoted to gateway role, while maximizing the throughput of the network in the saturation operating condition. It is assumed that an ideal top-level channel among the gateways is present, so that the probability of packet loss from the gateways and the central station of the WSN (or for any other network organization at the gateway level) is negligible. The problem of selecting *k* boards among the available *n*, leads to the following search space.

$$\sum_{k=1}^{k=n}\binom{n}{k} = 2^n \quad \text{(for the Newton's theorem)}$$

That, being exponential, makes it hard discovering the optimal solution with realistic values of *n*. To overcome such obstacle, we propose four algorithms, whose range of applicability depends on the cardinality (size) of the WSN. Quantitative comparisons are reported in Section VII.

1) Full Search

In this case we enumerate all possible solutions, selecting only the subsets which are Pareto-efficient. This approach is affordable only for WSNs composed of few boards (less than 10), namely with some tenths of sensors, and the optimal solution can be found.

2) Exponential search

Given the number of gateways *k*, we consider all the corresponding solutions that are Pareto-efficient. Since we do not consider all the possible cardinalities of gateways, the output can be sub-optimal. The algorithm, whose details are omitted due to the lack of space, considers a properly generated exponential ramp of the possible values of the *k* gateways. In our test cases (see Section VII), this strategy most of the times lead to the optimum.

3) K-Clusters (modified)

This strategy sits on top of the K-means clustering algorithm [21], as recalled below. To improve convergence towards near-optimal solutions, we introduced some constraints coming from the application environments. In particular it is assumed that the probability of incurring in an error when the nodes and the gateways are communicating is exponentially dependent on power (F1), as it is usual in fairly open environments. The signal power F1 received during a transmission is a function of the distance *d* between the boards: $F1=1/d^{\alpha}$, with α typically in the range [2..3]. As a consequence, the probability F2 that a packet is not correctly received it is function of the received power as $F2=e^{-F1}$.

By exploiting this model of the channel between gateways and boards, we assume that two boards are incompatible if their error probability F2 is above a certain threshold. Based on this definition, we compute an incompatibility matrix M, flagging all the pairs of nodes suspected to be critical, when belonging to the same subnet.

The algorithm begins considering the starting set of gateways with cardinality k, computed calculating the incompatibility matrix M for a given threshold, then extracting the set of boards B from the set of all the pairs of incompatible boards from M and finally by selecting randomly a subset of the boards B, having cardinality k. In summary the strategy we propose, starting from the above defined set of gateways, iterates the following steps until the set of gateways changes:

- compute the subnets of the set of gateways;
- compute the centroid point of each subnet;
- take as new k gateways the set of k boards closest to the set of k centroids (one board for each centroid);
- if the set of gateways is changed, repeat the algorithm.

The distance between two boards is their communication error probability (F2). The information carried by M allows overcoming the possible criticalities of the K-means clustering.

4) K-Clusters (random)

This algorithm is the same as the one described above, with the difference that the starting point is randomly chosen, namely the initial k gateways are selected using a uniform probability distribution over the available boards. The sub-optimal solutions so identified, in our experiments are nearly always worse than the improvement proposed before.

5) Comparison

As confirmed by the experiments, full search is an important reference, but it cannot scale-up. Exponential search can be a valuable improvement up to medium size applications (tenths of sensors and boards); in fact, both full search and exponential search have exponential complexity. The difference between the Random and Modified K-Clustering is mainly in terms of the selection of the starting point, which in the former case is random while in the latter exploits the information of the Incompatibility Matrix. The complexity of both strategies is polynomial, but the goodness of the solutions discovered by the Modified Clustering, increases as the threshold get higher. Note that the threshold decreases when the density of the WSN become significant. The border case is when the boards are close to each other and constitute a single group; in such a case the communication error probability is very low for each pair of boards, and the choice of the gateway among the boards is almost random since, to this purpose, they are pretty equivalent.

The matter is different for more realistic large size WSNs, where the density is still high, but (depending on the sensing goal) the set of boards is well spaced and clusterizable in groups, so that the error communication probability among the groups (based on their distance) it is no longer negligible. In such a case the Modified Clustering significantly overperforms the Random version.

B. *Identification of the Sub-Nets*

The baseline for partitioning the WSN into subnets is the channel model defined in Section VI.A.3. The rationale is that each board has to belong to the subnet (associated to the gateway) for which it is minimum the probability of having communication errors, as described in Figure 2.

```
Input: Si: A solution already calculated, L
Output: subnets
foreach board Bx in L {
        if (Bx not in Si) {
                foreach board Gj in Si {
                        compute ErrorProb(Gj,Bx);}
set Bx in the subnet y such that
ErrorProb(Gy,Bx) is the min over the
different gateways Gj in Si;
}       };
```

Figure 2. Grouping of boards in sub-nets assigned to the gateways.

VII. EXPERIMENTAL VALIDATION

The methodology and the algorithms have been verified using both synthetic applications and some real use cases extracted from running research projects [12][13]. This section initially reports some results concerning the aggregation of sensors in boards. Then, it is shown how it is possible to scale-up for wider WSNs considering a hierarchical organization, even with subnets adopting different board-to-gateway communication protocols.

A. *Board-Level Optimization*

To highlight the importance of a quantitative tradeoff when moving toward realistic deployments, let us consider an application setup with the following sensing goal:

```
SG = Pressure(0,0,0)  AND (Temperature(0,0,0) < 30)
     AND Water(1,1,0) AND (Temperature(1,1,0) > 20)
     AND Water(2,2,0) AND (Temperature(2,2,0) > 20)
```

For the sake of clarity, the environment is open space, the sensor model is ideal, the time window is set to 1 second and there are no taboos specified. The other parameters of the cost are Const=12.5, K=0.5 and all the sensors have the same SCj=1, no matter their type. Markup and NRE are set to 0. The output of SWORDFISH is a set of 7 sensors (Tab.III).

TABLE III. OPTIMAL POSITION AND TYPE OF SENSORS.

Sensor	Position (x,y,z)	Type
S0	0,0,0	Pressure
S1	0,1,0	Temperature
S2	1,1,0	Water
S3	2,1,0	Water
S4	2,2,0	Temperature
S5	1,0,0	Pressure
S6	3,2,0	Water

Starting from this configuration, PESCA computes the following two cliques with the max cardinality: {so, s1, s2, s3, s4, s5}, {s4, s6}. Then the covering of G' is performed using the subgraphs: {s0,s1,s2,s3,s4,s5} and {s6}. Due to space limits, the entire set of solutions generated and evaluated is not reported. In this example there is only a single solution in the Pareto frontier, which is the following.

```
Sol_1.  Board0={s0,s1,s2,s3,s5}, position=(1,1,0),
        Board1={s4, s6}, position=(2,2,0),
        Total cost=34.8, hardness=41
```

In the case we consider a different technology with Const=2.0 instead of 12.5, the solutions populating the Pareto frontier become those depicted in Tab.IV.

TABLE IV. PARETO SOLUTIONS FOR CONST=2.

	Sol_2	Sol_3	Sol_4
Board0	{s0,s5} (0,0,0)	{s1,s2,s3} (1,1,0)	{s4,s6} (2,2,0)
Board1	{s0,s5} (0,0,0)	{s1,s2} (1,1,0)	{s3,s4,s6} (2,2,0)
Board2	{s0,s5} (0,0,0)	{s1,s2} (1,1,0)	{s4,s6} (2,2,0)
Board3	--	--	{s3} (2,1,0)
Tot. Cost	**14.8**	**14.8**	**16.5**
Hardn.	**78**	**78**	**61**

It worth nothing that these solutions require more boards w.r.t. the previous one, as a consequence of the reduction of the board model fixed cost. Concerning the "quality" in terms of performance, the hardness (badness) of all the solutions is better (lower) than Sol_1. This behavior is reasonable, since the more board are used, the closer to the optimal output of SWORDFISH are the sensors.

In Figure 3. it is reported the analysis carried out on a more complex WSN: SG of 16 basic predicates producing an optimal network with 18 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. It is possible to observe the impact on performance and cost of the spreading vs clustering of sensors: the proposed quantitative analysis produces a significant value added for the designer when the tradeoff is not so clear: the driver may be not only the cost, but also the capability of the WSN to fulfill the initial application requirements.

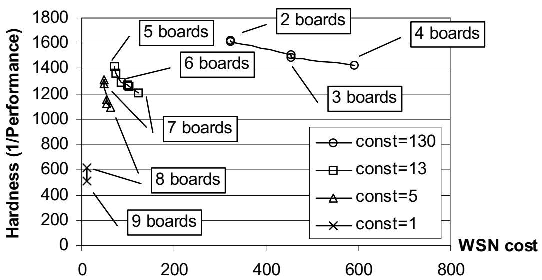

Figure 3. Pareto frontier of cost-performance analysis.

B. *Sub-Net Optimization*

In the following we reported some use cases derived from research projects [12][13], summarized in Tab.V. We also stressed the tools and methodology considering synthetic networks with sensing goals composed of over 16 predicates, tenths of boards and hundreds of sensors, confirming the results in terms of network organization and computation runtimes, such those of the third case of Tab.V

TABLE V. SOME OF THE CONSIDERED USE CASES.

	# Pred.	#Sensors	#Boards	Search Space (Full)
1	5	20	8	255
2	5	30	15	27 243
3	5	45	33	1434 119

Concerning the comparison among protocols, the key approximations adopted in the implementation of the MAC models of Section VI are those adopted in [12] and [13]. In addition, there is no interference among the sub-nets assigned to different gateways and the top level channel connecting the gateways is considered ideal. For the 802.11 protocol it is assumed a constant and independent collision probability of a packet transmission by each node, regardless the number of retransmission already suffered. To the purpose of throughput analysis, (conservative) saturation conditions are assumed, i.e. the transmission queue of each node is supposed to be always non empty. As far 802.15.4 is concerned, similar assumptions hold, with some tuning on the backoff counter value. More details about of the adopted protocol models can be found in [19][20].

The experimental data have been analyzed in order to compare the proposed strategies in terms of overall network organization, spanning of the search space and computational resource requirements. Tab.VI summarizes the results regarding the three use cases of Tab.V, considering also the impact of different communication protocols for each subnet assigned to a gateway. We highlight the search space, the solution discovered by each algorithm and which part of them is Pareto-efficient.

TABLE VI. POPULATION OF THE SOLUTIONS.

		Considered			Found			Pareto-eff.		
Alg.	**MAC**	**1**	**2**	**3**	**1**	**2**	**3**	**1**	**2**	**3**
Full S.	802.11 rts/cts	255	--	--	62	--	--	62	--	--
Full S.	802.11 base	255	--	--	40	--	--	40	--	--
Full S.	802.15.4	255	--	--	36	--	--	36	--	--
Exp S.	802.11 rts/cts	114	12 K	--	42	4 K	--	42	4 K	--
Exp S.	802.11 base	114	25 K	--	22	2 K	--	22	2 K	--
Exp S.	802.15.4	232	15	--	56	6 K	--	32	6 K	--
Modif.	802.11 rts/cts	8	16	33	7	14	28	6	14	28
Modif.	802.11 base	8	15	33	7	14	28	6	14	28
Modif.	802.15.4	8	15	33	6	14	27	5	14	27
Rand.	802.11 rts/cts	8	15	33	6	14	16	1	3	4
Rand.	802.11 base	8	15	32	5	12	12	1	3	4
Rand.	802.15.4	8	15	32	2	8	13	0	1	2

First use case of Tab.VI points out that:

- the Modified K-Cluster algorithm discovers a number of Pareto efficient solutions over its solutions found, that is close to 100%;
- the percentage of Pareto efficient solutions over the number of solutions, found by the Random k-Cluster algorithm, is around 20%.

The K-Cluster Modified algorithm behaves much better than the Random version thanks to the exploitation of domain-specific information (the incompatibility matrix).

In the second use case, the solution is not ensured to be the optimal one, since a comparative full search requires too many resources to be computed: 27243 solutions should be considered. Gray entries in Tab.VI correspond to the cases whose complexity needs runtimes over tenths of minutes, with main memory requirements over the GByte. However, the solutions discovered using the other strategies are Pareto efficient over the solutions set considered by the heuristics.

The results of the third analysis refer to a rather big WSN. Still in this case, any comparison of the heuristics with the full search (1434119 solutions) and the exponential algorithm is impossible. Our improvement to the K-Cluster algorithm always found better solutions w.r.t. the random choice.

Finally, we observed that the selection of the protocol impacts on the throughput of the subnets. Since this is one of the optimization objectives, the adopted standard influences the belonging of a solution to the Pareto frontier. In our tests, the 802.11 base protocol produces architectures with more subnets (gateways) than the others, because of the throughput tends to decrease significantly, as the number of nodes of a subnet increases. Such behaviour is manifest more smoothly with the 802.15.4 standard.

We also compared more in detail the different protocols by varying the number of boards and the average throughput for a broader benchmark set, under the traffic assumption of Section VI, and also in the case of a fair allocation of the communication channel [22]. The plots, not reported here due to space limits, clearly identify the different behaviour of the two protocols for WSN applications. When the number of boards becomes significant (above the dozen), fair protocols such as 802.15.4 which uses CSMA/CCA [22] are better in performance. For smaller numbers of boards, not fair protocols like 802.11 are the best in performance. It is also evident the importance of a quantitative evaluation, when the hardware cost of the protocol implementation is different. For numbers of boards around the dozen the cost of the protocol chipset can influence the selection of the optimal protocol.

VIII. Conclusions

The paper outlined the design flow implemented in SWORDFISH, focusing mainly on the problem to tradeoff cost effectiveness and capability to recognize the events. The proposed approach allows the user to identify a placement and clustering of sensors into realistic boards, optimizing the realization and deployment costs. Acceptable performance degradation is also ensured quantitatively. The toolset is also capable to deal with more complex goals, such as:

- Clustering of the optimal set of sensors using a pre-defined set of sensor boards, to mimic the adoption of off-the-shelf solutions (not detailed in this paper);
- Scaling of the WSN complexity, to cover also applications where it is mandatory to identify subnets assigned to gateways.
- Comparison of alternative protocols for each subnet assigned to a gateway.

Suitable exploration algorithms have been proposed, enabling the methodology to scale-up with the complexity of the problem, in acceptable analysis runtimes.

Note that the presented approach is complementary to the typical simulation-based analysis frameworks, since its emphasis is more on the system-level steps of the design, where a broad design space has to be extensively and efficiently explored, and on the formal modeling and verification of the WSN objectives and cost optimization. Moreover, as outlined in [14], SWORDFISH enables also sensitivity analysis of the performance against modification of the sensors position, observation windows, sensor accuracy, etc.

Work is in progress as part of large multi-partners projects [12][13]. Our methodology and toolset, which have been also stressed with more complicated synthetic use cases w.r.t. those of the projects, produces overall system organizations that are similar in structure (but better in performance and cost) to those adopted in practice: boards with 3-7 sensors and gateways every few boards (tenths of sensors).

Current effort is on the semi-automatic generation of application code, considering also energy-related constraints and robustness of the overall WSN in the case of fault and graceful degradation (ageing) of sensor accuracy.

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] K.Chakrabarty, S.Iyengar, H.Qi, E.Cho. Grid coverage for sourvelliance and target location in distributed sensor networks. IEEE Trans. on Computers, vol.51, Dec. 2002. pp 1448-1453.

[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] N.Bulusu, J.Heidemann, D.Estgrin. Adaptive beacon placement. Proc. of Int. Conf, on Distributed Computing Systems, Phoneix, April 2001. pp 489-498.

[6] Y.Zou, K.Chakrabarty. Sensor Deployment and target localization based on virtual forces. Proc. IEEE Infocom Conf, vol.2, pp 1293-1303, San Francisco, CA, April 2003.

[7] K. Kar, S. Banerjee, Node placement for connected coverage in sensor networks, Proc. of Int. Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks, 2003.

[8] 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.

[9] 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.

[10] 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.

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

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

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

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

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

[16] Chipcon, Data Sheet of CC2430 2.4 GHz IEEE 802.15.4 / ZigBee RF Transceiver. http://www.chipcon.com, 2006.

[17] 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.

[18] A.W.Appel, Modern Compiler Implementation in Java, Cambridge University Press, 1998. pp 195-209.

[19] G.Bianchi. Performance analysis of the IEEE 802.11 distributed coordination function. Proc. of IEEE Journal on Selected Areas in Comm., vol.18, March 2000. pp535-547.

[20] T.Park, T.Kim, J.Choi, W.Kwon. Throughput and energy consumption analysis of IEEE 802.15.4 Slotted CSMA/CD. In Electronics Letters, vol. 41, Sept. 2005.

[21] T.Kanungo, D.Mount, N.Netanyahu, C.Piatko, R.Silverman, A.Wu, An Efficicent k-Means Clustering Algorithm: Analysis and Implementation, IEEE Trans. on Pattern Analysis and Machine Intelligence. Vol.24 n.7, July 2002. pp 881-892.

[22] X.Wang, G.B.Giannakis, CSMA/CC: A modified CSMA/CA Protocol Mitigation for the Fairness Problem for IEEE 802.11 DCF. EURASIP Journal on Wireless Communications and Networking, v.2006 n.2, p.40-40, April 2006.