

# The Dark Side Of DEMMON: What Is Behind The Scene In Engineering Large-Scale Wireless Sensor Networks

Stefano Tennina<sup>\*</sup>, Ricardo Gomes and  
Mario Alves  
CISTER Research Center  
Polytechnic Institute of Porto (ISEP/IPP)  
Porto, Portugal  
sota@isep.ipp.pt, rftg@isep.ipp.pt,  
mjf@isep.ipp.pt

Vincenzo Ciriello and Gabriella Carrozza  
SESM scarl  
Giugliano in Campania  
Naples, Italy  
vciriello@sesm.it, gcarrozza@sesm.it

## ABSTRACT

Most research work on WSNs has focused on protocols or on specific applications. There is a clear lack of easy/ready-to-use WSN technologies and tools for planning, implementing, testing and commissioning WSN systems in an integrated fashion. While there exists a plethora of papers about network planning and deployment methodologies, to the best of our knowledge none of them helps the designer to match coverage requirements with network performance evaluation. In this paper we aim at filling this gap by presenting an unified toolset, i.e., a framework able to provide a global picture of the system, from the network deployment planning to system test and validation. This toolset has been designed to back up the EMMON WSN system architecture for large-scale, dense, real-time embedded monitoring. It includes network deployment planning, worst-case analysis and dimensioning, protocol simulation and automatic remote programming and hardware testing tools. This toolset has been paramount to validate the system architecture through DEMMON1, the first EMMON demonstrator, i.e., a 300+ node test-bed, which is, to the best of our knowledge, the largest single-site WSN test-bed in Europe to date.

## Categories and Subject Descriptors

B.4.4 [Performance Analysis and Design Aids]: [Simulation, Verification, Worst-case analysis]

## General Terms

Design, Experimentation

## Keywords

Large Scale Wireless Sensor Networks, Toolset, Testbed

<sup>\*</sup>Corresponding author.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

MSWiM'11, October 31–November 4, 2011, Miami, Florida, USA.  
Copyright 2011 ACM 978-1-4503-0898-4/11/10 ...\$10.00.

## 1. INTRODUCTION

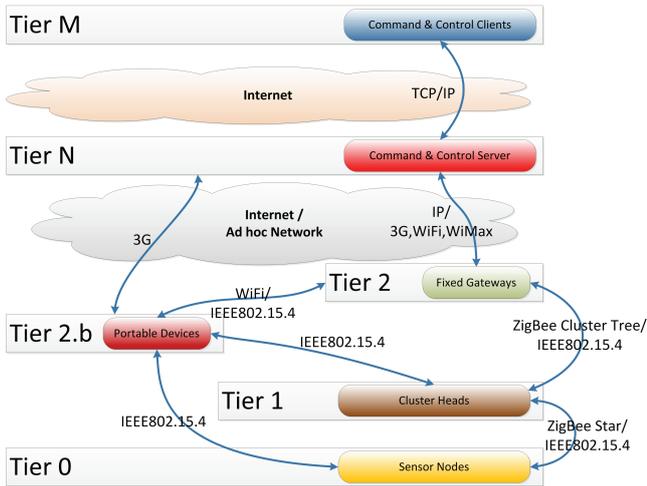
Wireless Sensor Networks (WSNs) have emerged as an infrastructure to support new classes of large-scale and dense networked embedded systems. While in the last decade there has been a plethora of scientific publications on WSNs, the vast majority focuses on protocols and algorithms (e.g. medium access control, routing, data aggregation) while only a few papers report on real(istic) applications [24].

Several relevant work on WSN architectures and solutions supported by working prototypes have been described in the literature (e.g. [5, 12, 14, 19]) and in the scope of research projects (e.g. [2, 11, 23]). However, in [28] we have shown that none of them fulfills all requirements for large-scale and dense real-time monitoring.

EMMON [1] is an ARTEMIS industry-driven project that aimed to design a WSN system architecture targeting large-scale and dense real-time monitoring applications, such as real-time pollution/noise maps of a city, or critical infrastructures monitoring (such as bridges, tunnels or the power grid). The EMMON architecture encompasses all system components: command and control graphical user interface, communication network architecture, middleware and hardware platform. It is (as far as possible) built upon standard and commercially available (COTS) technologies, maintaining as much flexibility as possible, but still guaranteeing its adequateness to fit specific applications' requirements [21]. We envisage to fulfill Quality-of-Service (QoS) requirements in an integrated fashion, considering scalability, timeliness, reliability/robustness and energy-efficiency. These QoS requirements can be found in applications [21] such as energy-efficiency in data centers [19] and infrastructures monitoring (e.g., bridges, tunnels or the power grid) [8].

Importantly, and as reported in this paper, this WSN architecture is supported by a *EM-set*, a unique and complete planning, dimensioning, simulation and analysis toolset, for deployment planning, worst-case dimensioning, protocol simulation, remote programming and network sniffing. This paper outlines such a toolset and shows how it has been used to test and validate the baseline EMMON architecture by extensive simulation and experimental evaluation, including through a 300+ node test-bed [1], which is, to the best of our knowledge, the largest single-site WSN test-bed in Europe.

The remainder of the paper is organized as follows. After a brief overview of the EMMON system architecture in Section 2, the focus is put on the EM-set, encompassing network



**Figure 1: EMMON Hierarchical System Architecture.**

deployment planning (Section 3), analytical and simulation models (Section 4), nodes programming/testing and experimental data gathering (Section 5). Finally, Section 6 illustrates some results of the first instantiation of the EMMON architecture on a physical test-bed (DEMMON1<sup>1</sup>) and compares them to simulation and analytical results. Concluding remarks are given in Section 7, as well as an outline of the on-going work related to the instantiation of the EMMON system architecture into several application scenarios.

## 2. EMMON ARCHITECTURE & TOOLSET

This section outlines the EMMON architecture and the toolset used to support it and to provide the results described in Section 6.

### 2.1 System Architecture

Building on the alternatives identified in [28] and to cope with scalability and QoS requirements, EMMON adopts a hierarchical, multi-tier network architecture as sketched in Fig. 1. Its main characteristics are summarized as follows.

(i) The synchronized version of the IEEE 802.15.4 MAC is used at the lowest tiers. Nodes are synchronously active or sleeping, with a dynamically adaptable duty-cycle (each cluster can operate at different duty-cycles). This enables to find the best delay/throughput vs. energy trade-off. Both best-effort (CSMA/CA, during the CAP) and real-time (GTS, during the CFP) traffic classes are supported.

(ii) WSN nodes are organized into a ZigBee-based Cluster-Tree network model [17], rooted at a gateway playing the role of the sink. A cluster-tree is a hierarchical architecture *per-se* and is adequate for convergecast traffic, as utilized in the majority of WSN applications. However, to avoid collisions between clusters while meeting all end-to-end deadlines of a predefined set of time-bounded data flows and minimizing the energy consumptions of the nodes, clusters' active portions are scheduled in a non-overlapping sequence – using the Time Division Cluster Scheduling (TDCS) [17].

(iii) We adopt tree-routing for upstream traffic, which has a negligible memory footprint and processing delay since

<sup>1</sup>The acronym stands for EMMON Demonstrator, phase 1.

no routing tables are needed. We also support efficient geographical-based routing of queries for the downward flow, for disseminating requests from a single root to all the nodes involved. This has a huge impact in terms of scalability, since it allows users to interact with the system through the definition of geographical objects, rather than any explicit request for raw readings from specific sensor nodes.

(iv) Beacons inherently enables the support for time synchronization at the Data Link and Network layers. Accordingly, it enables accurate time stamping of sensor data (required by many applications), energy-efficiency through duty-cycling, cluster scheduling techniques and a contention-free MAC.

(v) Data aggregation, sensor- and data-fusion mechanisms [22] are implemented at all levels of the architecture: (v.a) at the sensor nodes (SNs), by aggregating multiple readings taken over time (temporal aggregation), before sending these data upstream; (v.b) at the cluster heads (CHs), by aggregating multiple readings coming from different sensors or children CHs (spatial aggregation), before forwarding the report upstream; (v.c) at the gateway (GW), where sensor fusion, i.e., inferring useful information through, e.g., model fitting, is done by considering multiple reports coming from the CHs, before sending them to the C&C; and (v.d) at the C&C, where a complete information can be returned to the users by allowing, e.g., a correlation of the incoming sensor reports with other available data (e.g., current traffic conditions in urban noise or air quality monitoring applications [21]).

(vi) A novel EMMON-specific middleware runs on all the elements of the system: it glues all the components together, from the C&C clients to the SNs, leading them to work properly over heterogeneous communication technologies (Fig. 1). It supports both periodic measurements reporting and event based (alarms) applications. It also greatly helps in networking and system management operations, thanks to its distribution of the intelligence as low as possible in the network's tiers.

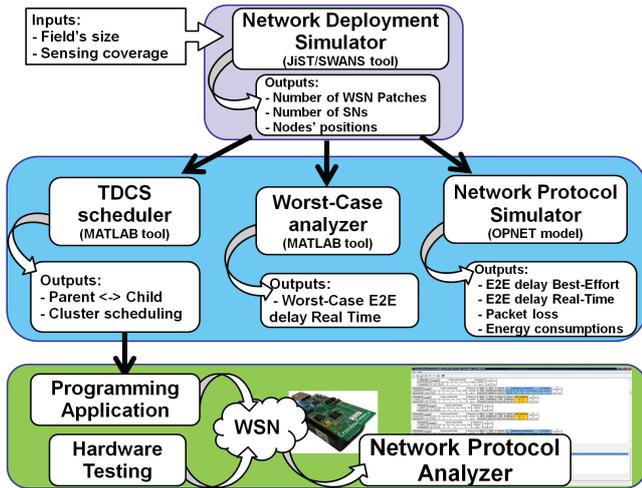
(vii) The EMMON C&C subsystem is the interface to the end-users. It is composed by two applications, one which runs on the C&C server and uses the middleware API, and the other on the C&C clients, where the Graphical User Interface is implemented.

Finally, the baseline IEEE 802.15.4 and ZigBee protocol stack is supported by a solid critical mass, designed and implemented in synergy with the TinyOS 15.4 and ZigBee Working Groups [13].

### 2.2 EM-set: System Planning, Simulation, Analysis toolset

This section provides a snapshot of EM-set, the EMMON toolset, before describing each component in Sections 3 – 5.

Fig. 2 provides an overall perspective of EM-set. It is worthwhile to stress that the integration between the different components (inputs/outputs) into a single framework enables to speedup system design. In fact, the ultimate goal of this toolset is to thoroughly allow functional and non-functional performance indices evaluation under several aspects. For instance, this toolset allows to assess the scalability limits of the EMMON network through an evaluation of e.g., the end-to-end delays derived from analytical and simulation models as well as from experimental trials. The results presented in Section 6 derive from the use of *EM-set*.



**Figure 2: EM-set: EMMON toolset for network planning, dimensioning simulation analysis and test/programming.**

Starting from the field area and the sensing coverage of each SN and assuming the structure of a basic WSN Patch, i.e., a GW and  $C$  CHs surrounding it, the Network Deployment Planning outputs the number of WSN Patches and SNs needed to cover that area, as well as their optimal positions. These outputs feed the TDCS scheduler, the Worst-Case analyzer and the Network Protocol simulator. These three tools enable network dimensioning and a performance evaluation. Upper bounds on the end-to-end delays of real-time traffic and end-to-end delays for both best-effort and real-time traffic classes, packet losses and network lifetime (as per node energy consumptions) are computed. The TDCS scheduler outputs the topology of the WSN Patch and the clusters' scheduling that are used to feed the Remote Programming and Testing tools (to program/test the WSN nodes via a USB tree) and to evaluate the network performance through Network Protocol Analyzers (i.e., a sniffer tool to capture the packets and a customized parser to extract the most useful statistics from the logs).

### 3. NETWORK DEPLOYMENT PLANNING

The Network Deployment Planning is composed by a single engine: a simulator to study the optimal deployment problem. Assuming that in EMMON there is some control over node deployment, i.e., CHs and GWs are placed in order to maximize network connectivity, the SNs are generically supposed to be scattered all over the monitored area. We built upon an Open Source tool named SiDNet-SWANS [25], built over JiST, a java based discrete event simulator. We choose JiST/SWANS because it allows to easily simulate very large scale WSNs. The results in [29] confirm that JiST/SWANS presents better time execution performance, compared to more recent simulators as NS-3 or OmNet++. Moreover, [6] shows that JiST/SWANS is really capable to simulate networks composed by a high number of nodes (up to one million) in acceptable time and using common PCs. NS-3 and OMNeT++ experiments showing similar capabilities are not available. [26] confirms the qualitative features

of the JiST's simulation, comparing it with NS-2 and getting very similar performance results.

In our scenario, the deployment problem is composed by two levels: (i) inter-patch deployment, i.e., how to distribute the WSN patches on the field, and (ii) intra-patch deployment, i.e., how to place the nodes internally to the patch, in particular how to arrange the CHs and the GW. The designed tool is based on innovative deployment strategies. The inter-patch deployment organizes the field as a grid, where each cell has a specific geometric shape (e.g., square, triangular or hexagonal). This deployment strategy has already been discussed in several studies [7, 31], but we use it in a slightly different way. In those approaches, the nodes (or a set of nodes in some cases) are placed on the vertices of each cell (e.g., on the corners of the square, or on the vertices of triangles), while, in our approach, the nodes are placed inside each cell, meaning that each cell contains a WSN Patch having the GW in the middle. Currently, the designed tool provides two alternative grid deployments: square based deployment (SBD), where each WSN patch is circumscribed in a square and triangle based deployment (TBD) in which the patches are inscribed into equilateral triangles. Fig. 3 shows WSN Patches deployed in e.g., a square-shaped grid.

As far as the intra-patch deployment is concerned, the objective is to organize the deployment based on a Cluster-Tree model, as shown in Fig. 3: the diamond markers are the GWs<sup>2</sup>, while the star markers represent the CHs<sup>3</sup>. The WSN Patch is logically organized according to a tree based topology, in which the GW acts as the root and each CH as node. Hence, according to the EMMON assumptions (i.e., to have some control on the GW and CHs positions) the designed deployment scheme provides the positions of CHs organized in "rings" around the GW (for all patches), while SNs (point markers in Fig. 3) are randomly scattered over the entire field. Then, nodes must organize themselves into a Cluster-Tree topology, using the IEEE 802.15.4 standard association mechanism, i.e., each node chooses its parent based on some metrics (e.g., the best link quality level). As a consequence, the deployment planning tool actually runs network simulations taking into account realistic signal propagation conditions. However, to allow for fault tolerance and recovery strategies to be implemented in the EMMON system, an additional requirement has been included, i.e., that each child should have at least two alternative candidate parents to choose from. This imposes a certain degree of redundancy on the number of CHs.

Finally, the deployment schemes are evaluated against two performance indices: (i) network connectivity and (ii) field's coverage. Connectivity refers to how much of the generated data can ultimately arrive at the C&C, from the SNs and through the CHs and the GWs. A network is considered to be fully connected if all nodes can deliver their data to the destination. According to the definition reported in [30, 32], a field is considered to be covered if all its points are within the sensing range of at least one active sensor. The sensing range  $R_r$  is defined as the maximum distance at which a sensor node can measure environmental parameters (e.g., temperature, humidity, light, etc.) and is an input to this tool. In EMMON, we are interested in checking both network connectivity and field's coverage simultaneously. Therefore we

<sup>2</sup>Every GW is placed in the center of each cell.

<sup>3</sup>Every CH surrounds a GW in each cell.

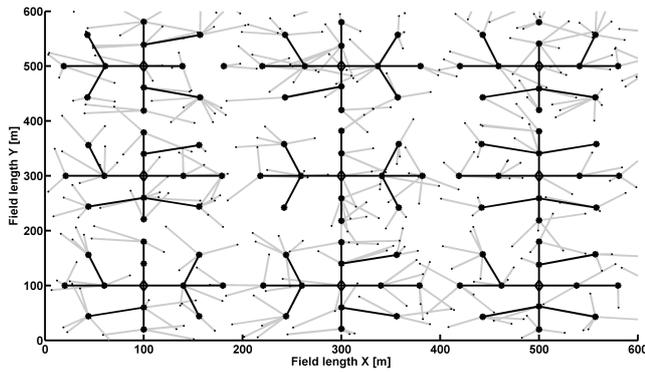


Figure 3: Example of WSN Patches with a SBD scheme to cover at least 80% of a 49 km<sup>2</sup> field (particular). Diamond markers are GWs; stars are CHs, distributed along 2 hops around each GW; points are SNs, randomly deployed.

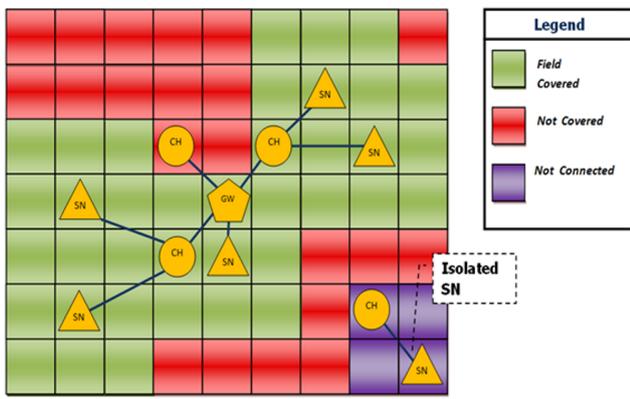


Figure 4: Field coverage analysis example.

make the latter definition more restrictive by assuming that a point is *covered* if there is at least one SN less than  $R_r$  meters away from it and that SN is connected to the network, i.e., it is able to deliver its measurements to the C&C.

Fig. 4 reports an example of this concept: the triangles, circles and pentagon represent SNs, CHs and GW, respectively, while the blue lines represent the link between the nodes. For the sake of simplicity, for this example, let us assume that the connection between the GW and the remote C&C is guaranteed at any time. So, if a SN has a path to the GW, we say that it is connected. Furthermore, let us suppose that each SN has a sensing range equal to one cell, i.e., each SN can measure environmental data only from its eight adjacent cells. Then, the green cells in Fig. 4 represent all the points covered: there is at least a SN in an adjacent cell, and the SNs covering the cell are connected. Both the red and violet cells are uncovered: the former, because they are not close to any SN; the latter, because a close SN does not have any path to the GW.

The field coverage analysis algorithm is reported in Fig. 5. It is implemented as a post-processing analysis tool and requires as inputs the results of the simulation. It is then able to output the field coverage analysis. In particular, the algorithm requires a list  $L_s$  of all the nodes positions and

#### • Input:

- $L_s$ : list of sensor positions on the field.
- $R_r$ : sensor reading range.

#### • Algorithm:

- Field is represented as a boolean matrix  $F$ .
- $\forall(x, y) \in L_s$  considering a square  $A$  of side  $2R_r$  centred in  $(x, y)$ .
- $\forall(x_i, y_j) \in L_s$ :

$$\text{if } (\sqrt{(x-x_i)^2 + (y-y_i)^2} < R_r) \Rightarrow F[x_i, y_j] = 1$$

- **Field coverage** =  $\frac{\text{\# elements of F equal to 1}}{\text{\#field's point}}$

Figure 5: Field coverage analysis algorithm.

considers the sensing range  $R_r$  as a parameter. The field is represented as a boolean matrix where each element denotes if the corresponding cell is covered or not. To this aim, we should consider as covered those cells that fall into a circle centered on the SN and whose radius is equal to  $R_r$ . The field coverage is then calculated as the ratio between the covered cells and the whole area.

## 4. PERFORMANCE ANALYSIS

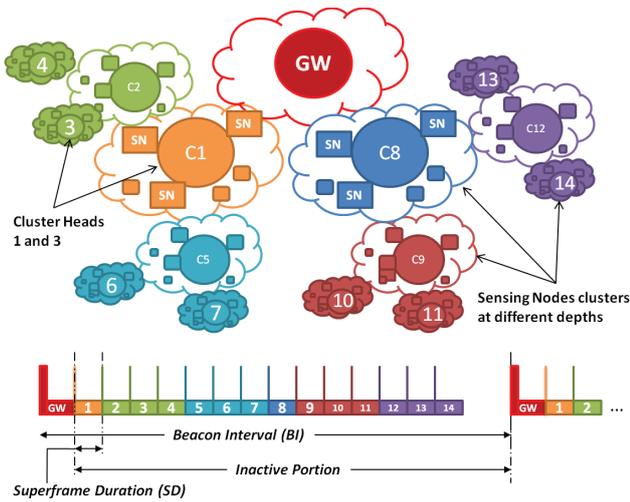
Focusing on the portion of the EMMON architecture below the GW, i.e., on a *single* WSN Patch, the outputs from the network deployment simulator are translated into appropriate inputs for three tools [15]: the TDCS Scheduler, the Worst-Case Analyzer and the Network Protocol Simulator (recall Fig. 2). These tools are applied to each WSN Patch, as outlined next.

### 4.1 TDCS Scheduler

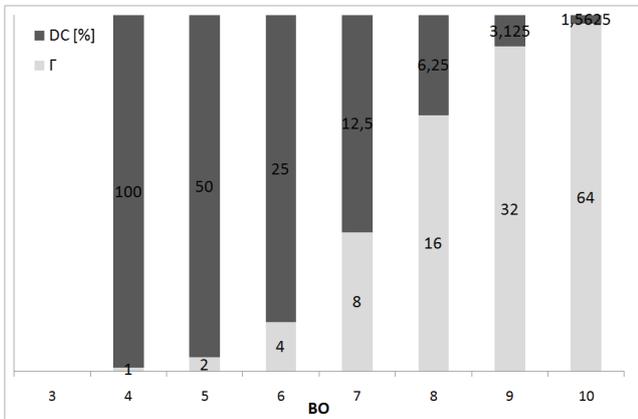
In the Cluster-Tree WSN model, clusters' activity is scheduled according to a time division approach, where clusters' active periods do not overlap. Beacons are messages sent by every local coordinator of any WSN Patch (i.e., the GW and the CHs) and serve to maintain the synchronization among the nodes of each cluster. This has the advantage of improving the coordination to save energy (reduce retransmissions, put the nodes to sleep and wake them up again in a synchronous fashion) and of guaranteeing a given level of QoS. However, to preserve the coordination and avoid intra-cluster collisions, the TDCS algorithm [17] is needed. This mechanism involves the definition of the Start Time values of the MAC protocol, such that the active portions of each cluster are interleaved during the inactive portion of all the others sharing the same collision domain, as shown in Fig. 6.

This design choice leads to a given upper bound on the allowed number of clusters ( $\Gamma^4$ ) and related duty cycles ( $DC$ ). Knowing that  $0 \leq SO \leq BO \leq 14$  (being SO the Superframe Order and BO the Beacon Order defined in the IEEE 802.15.4 standard), the maximum number of allowed clusters is  $\Gamma = 2^{BO-SO}$  and the associated duty cycle is  $DC = 1/\Gamma = 2^{SO-BO}$ . An example of these relations is shown in Fig. 7 where SO has been fixed to 4. As a consequence, under the assumption that the couple (BO,SO) is

<sup>4</sup>The total number of clusters in a WSN Patch must include also the GW, as a special cluster head.



**Figure 6: Time Division Cluster Scheduling (TDCS):** it is assumed that every cluster has the same value of the couple (BO, SO). The inactive portion of the GW (only) is also shown, as an example.



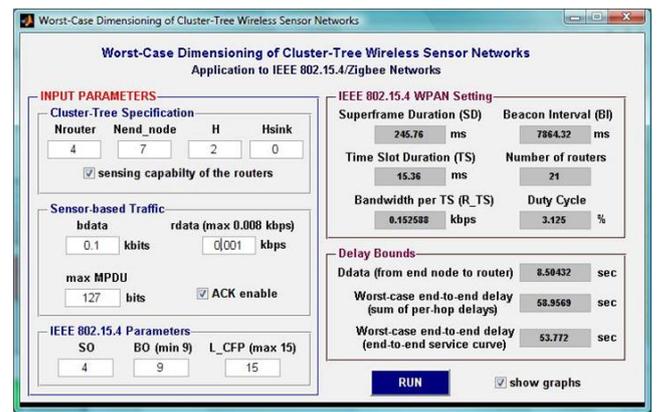
**Figure 7: TDCS: upper bound on the allowed number of clusters ( $\Gamma$ ) and related duty cycles (DC) for  $SO=4$  and a subset of values of BO.**

the same for every cluster in a WSN Patch, at design time, those relations help identifying the best trade-off between  $\Gamma$  and the related DC.

The TDCS Scheduler implements this mechanism. Given the WSN Patch characteristics (i.e., parent-child relation in the Cluster-Tree scheme), this tool (built in MATLAB) computes the minimum BO along with the start time for each cluster head to schedule its active portion (superframe), as shown in Fig. 6.

## 4.2 Worst-Case Analyzer

The Worst-Case Analyzer is a MATLAB tool (Fig. 8) which estimates an upper bound for the end-to-end delay of the real-time traffic in IEEE 802.15.4/ZigBee Cluster-Tree Wireless Sensor Networks, i.e., the traffic whose packets are sent during the contention free portion (GTS slots) of the superframe.



**Figure 8: MATLAB tool for worst-case analysis and dimensioning.**

This tool builds on the Network Calculus mathematical model, as described in [17] and enables to iteratively find the best duty-cycle vs. delay/throughput trade-off, by varying network parameters such as sensor traffic.

## 4.3 Network Protocol Simulator

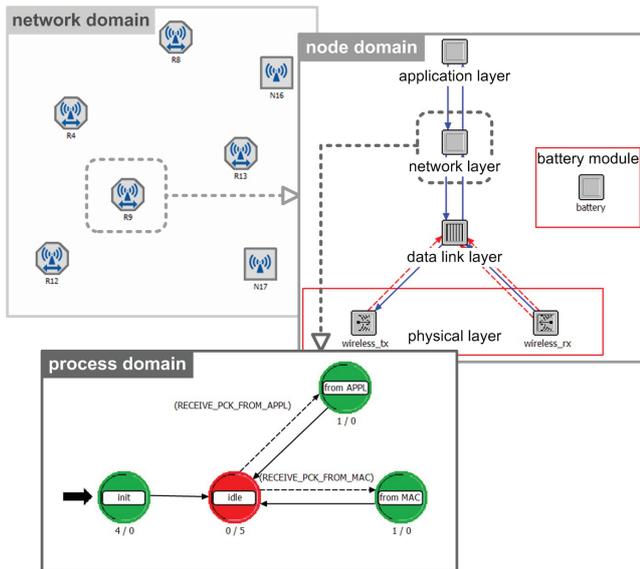
The performance of the communication protocol is accurately evaluated using a simulator built in OPNET [3]. This is a complete simulation model for IEEE 802.15.4/ZigBee that complements a previous model [16].

Recently, several analytical and simulation models of the IEEE 802.15.4 protocol have been proposed. Nevertheless, currently available simulation models [27] for this protocol are both inaccurate and incomplete, and in particular they do not support the GTS mechanism, which is required for time-sensitive wireless sensor applications. Although a commercial software (but free for academia), OPNET Modeler was chosen due to its accuracy and to its sophisticated graphical user interface.

The idea behind this simulation model was triggered by the need to build a very reliable model of the IEEE 802.15.4 and ZigBee protocols for Wireless Sensor Networks (WSNs). The IEEE 802.15.4/ZigBee simulation model builds on the wireless module, an add-on that extends the functionality of the OPNET Modeler with accurate modeling, simulation and analysis of wireless networks. According to the overview presented in Section 2, three types of nodes are implemented, namely a GW, a CH and a SN. All types of nodes have the same internal architecture (node domain), but they differ in the available user-defined attributes.

The structure of the IEEE 802.15.4/ZigBee simulation model is presented in Fig. 9. The *physical layer* consists of a wireless radio transmitter and receiver compliant to the IEEE 802.15.4 standard running at 2.4 GHz frequency band with 250 kbps data rate. Default TelosB-related settings are used for the physical characteristics of the radio channel such as background noise and interference, propagation delay, antenna gain, and bit error rate.

The *data link layer* supports the beacon-enabled mode and implements two medium access control protocols according to the IEEE 802.15.4 standard, namely the contention-based slotted CSMA/CA and contention-free GTS. MAC payload (MSDU) incoming from the network layer is wrapped in MAC header and MAC footer and stored into two sep-



**Figure 9: The structure of the IEEE 802.15.4/ZigBee Opet simulation model.**

arate FIFO buffers, namely a buffer for best-effort data frames and another buffer for real-time data frames. The frames are dispatched to the network when the corresponding CAP or CFP is active. On the other hand, the frame (MPDU) incoming from the physical layer is unwrapped and passed to the network layer for further processing. The data link layer also generates required commands (e.g., GTS allocation, deallocation and reallocation commands) and beacon frames when a node acts as GW or CH.

The *network layer* implements address-based tree routing according to the ZigBee specification. The frames are routed upward or downward along the cluster-tree topology according to the destination address by exploiting the hierarchical addressing scheme provided by ZigBee.

The *application layer* can generate unacknowledged and/or acknowledged best-effort and/or real-time data frames transmitted during CAP or CFP, respectively. There is also a battery module that computes the consumed and remaining energy levels. The default values of current draws are set to those of the widely-used MICAz [9] or TelosB [10] motes.

With the help of this simulator, we can infer (Table 1) end-to-end delays for both real-time and best-effort traffic, as well as compute network statistics such as packet loss, network throughput and lifetime, via per-node energy consumption estimation. In particular, this simulation model gets user-defined attributes (network topology and nodes parameters) through an XML file, which we automatically generate through an ad-hoc MATLAB script, which takes as inputs the outputs of the Network Deployment Simulator (i.e., nodes positions and parent-child relations, Section 3) and the windows offset defined by the TDCS scheduling tool (Section 4.1).

## 5. REMOTE PROGRAMMING & TESTING

Moving from the outputs of the TDCS tool, a set of appropriate MATLAB/Unix SHELL scripts are used to automatically generate header files (.h), compile the firmware of

the nodes with each individual configuration and load the compiled firmware in parallel into each node, through the USB tree. Interestingly, using this tool to program in parallel the nodes over the USB tree we were able to save 66% in programming time with respect to a classical serial programming. In particular, in the DEMMON1 deployment, it took around 30 minutes to serially program a patch composed by 100 nodes, while with this tool the programming time reduced to no more than 10-12 minutes.

Finally, with the help of sniffer devices like [18] and a custom-designed log parser, built in C++ and MATLAB, EMMON specific data from the IEEE 802.15.4 frames are extracted. The outputs of the sniffer/parser (e.g., average and max delay) can then be compared with the theoretical (i.e., worst-case) and simulation (i.e., average and max) end-to-end delays.

We also designed an application to automate the testing of the hardware (USB cabling/hubs and TelosB nodes) having an element as reference (e.g., a previously tested TelosB node or USB hub). This testing tool compares the behavior of the elements under tests with the reference one. It was fundamental to early identify: (i) 19 out of 300+ TelosB with faulty humidity sensors (i.e., not usable as SNs); (ii) 2 out of 50+ USB hubs broken and (iii) a whole set of (3 m length) USB cables not compliant with the specification given to our local supplier.

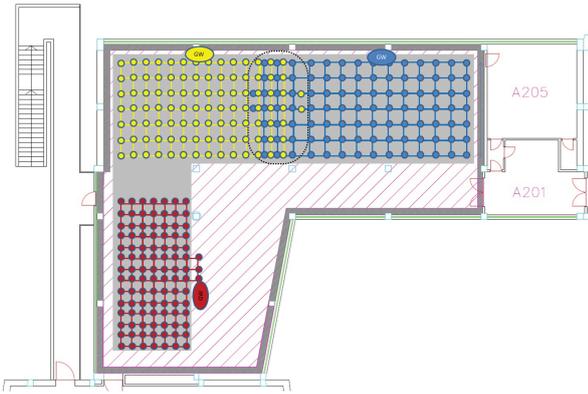
## 6. DEMMON1 CASE STUDY

This section illustrates DEMMON1, the first demonstrator of the EMMON architecture. A comparison between simulation, analytical and experimental results validates our EMMON architecture (in terms of the interoperability among its architectural components), characterizes its performance and scalability limits and shows the usefulness of the toolset presented in this paper.

### 6.1 Site Layout

To validate our architecture, we performed a real deployment as shown in Fig. 10(a). 303 TelosB nodes were organized into 3 WSN Patches, with the possibility of defining different topologies by programming the nodes over a USB tree using our toolset. The GWs communicated via wired LAN to a host PC running the C&C server. The WSN Patches simultaneously operated in three distinct frequency channels. These channels were chosen to minimize the impact of the actual external interference [20]. In order to do so, a pre-deployment Electro-Magnetic Interference (EMI) analysis in the deployment site has been performed using HeatMapper [4], a free Wi-Fi coverage mapping site survey tool made by Ekahau, whose results are shown in Fig. 10(b). In particular, Fig. 10(b) clearly shows that the WiFi access points all around the deployment site were using standard IEEE 802.11 channels (namely, ch.1, ch.6 and ch.11) allowing us to assign IEEE 802.15.4 ch.15, ch.25 and ch.26 channels to our three patches, as it was expected [20].

Each WSN Patch was composed of 100 nodes organized into a 3-hops Cluster Tree model having the GW as root, 3 CHs on the first hop and 9 on the second hop for a total of 12 clusters, each of them with 8 to 10 SNs. Since all nodes were sharing the same collision domain, i.e., nodes were able to hear/interfere each other, the CHs locations were randomly chosen but the TDCS tool was used to define a scheduling among the clusters in each WSN Patch. As already men-



(a) 300+ nodes divided into 3 WSN Patches. Patch1 (blue) and Patch2 (yellow) were partially overlapping.



(b) ISM spectrum (namely, WLAN) interference study in the site's area [4].

**Figure 10: DEMMON1 - First EMMON deployment (ISEP, Porto, Portugal).**

tioned, frequency division guaranteed no interference among the nodes of different WSN Patches.

To stress our architecture, we physically overlapped some nodes belonging to two patches (Fig. 10(a)). Any query cast from the C&C involving that area runs in parallel on those nodes and data aggregation is performed first at the CHs of each patch, then at the GWs and ultimately at the C&C level. This clearly demonstrate how distributing the intelligence on the lowest tiers as possible of the system is actually implemented.

## 6.2 Performance Analysis

Moving from the coverage analysis results, performance limits of the communication protocol have been evaluated by focusing on the WSN Patch level, i.e., the portion of the EMMON network below the gateway. In particular, using the previously referred toolset, several scenarios were identified based on all the combinations of the input parameters in Table 1. This setup generated different network topologies with maximum depth<sup>5</sup> ( $L_m$ ) ranging from 2 to 5 and

<sup>5</sup>Maximum number of hops between a SN and the GW.

**Table 1: WSN Patch Validation Setup**

Parameter	Range	Meaning
$R_m$	{2;3;4;5}	Number of children CH per parent
$\Gamma$	{5;17;21}	Number of Clusters
$\Sigma$	{5;10;15;20;24}	Number of children SNs per parent
$SO$	4	Superframe Order
$P$	137 B	NPDU size
$T$	2 s	Packet generation ratio

total number of nodes in one WSN Patch ranging from 25 to 501<sup>6</sup> (see Table 2).

In the simulated scenarios, the nodes' application layer generates and sends upstream a packet every  $T$  seconds. Packets have a fixed size of  $P$  bytes. Both *best-effort* (BE) and *real-time* (RT) traffic classes are generated with the same packet generation ratio (nodes use the contention-based access period and the contention free period of the IEEE 802.15.4 superframe, respectively). BE traffic is intended to simulate periodic sensor reports, while RT traffic accounts for event-driven alarm notifications. As a consequence, only CHs generate RT traffic: this reflects that only CHs should reliably trigger alarm notifications towards the GW, by filtering out noisy SNs readings.

The value of  $SO$  has been fixed to 4, which means an active portion of 960 symbols, while  $BO$  has been computed for each scenario by our TDCS MATLAB tool, as well as the set of CHs start times, in order to fit with the number of clusters ( $\Gamma$ ). We assume that every cluster in a WSN Patch has the same value of the couple ( $BO$ ,  $SO$ ), as in Fig. 6, and, recalling that the size of the collision domain is as large as the network size, i.e., nodes can interfere each other, the TDCS scheduler gives a conservative solution, where clusters' active portions do not overlap<sup>7</sup>. As a consequence, in our setup (Table 1),  $BO$  ranges between 7 and 9, corresponding to beacon interval (BI) values of  $\sim 2$  and  $\sim 8$  s, respectively, and duty cycles of 12.5% and 3.125%, respectively (Fig. 7).

In this paper, the end-to-end (e2e) delay for BE (e2e-BE) and RT (e2e-RT) traffic classes and the packet loss ratio are the performance indices, while other available figures (e.g., per-node energy consumptions) are not shown here, due to space constraints.

The same experiments were run live on our testbed. However, only best effort traffic was generated, using the CAP portion of the IEEE 802.15.4 MAC. The traffic was monitored through protocol analyzers with the help of some sniffer tools (like [18]) to compute the statistics for the e2e delay.

Table 2 shows an excerpt of the results related to network performance. In particular, a subset of all the scenarios is presented with an increasing level of network complexity, enabling the comparison between *simulation* and *experimental* results for the e2e-BE delay, as well as between *simulation*

<sup>6</sup>Remember that larger system scale can be achieved through the multiple patches mechanism.

<sup>7</sup>The tool is ready yet to cope with scenarios where two clusters are spatially apart to not interfere each other. In this case, their active portions can be overlapped, with an evident gain in terms of system delay, due to the reduction of the beacon interval.

**Table 2: Simulation, worst-case and experimental results (excerpt) from the campaign defined in Tab. 1.**

Tot Nodes	$\Gamma$ (clusters)	$\Sigma$ (SNs per CH)	$R_m$	$L_m$	BO	BI [s]	CAP - Best Effort						CFP - Real Time				
							Pkt Loss [%]	End-To-End Delay [s]						End-To-End Delay [s]			
								SIMULATION			EXPERIMENTAL			SIMULATION			ANALITICAL
								avg	std	max	avg	std	max	avg	std	max	WorstCase
25	5	5	4	2	7	2.048	0.56	1.73	1.12	3.73	1.32	0.36	2.06	1.95	1.13	3.92	5.82
45	5	10	4	2	7	2.048	2.39	1.80	1.12	3.80	1.59	0.40	1.99	1.97	1.13	3.91	5.82
65	5	15	4	2	7	2.048	5.53	1.81	1.12	3.82	1.67	0.38	2.14	1.97	1.14	3.91	5.82
85	5	20	4	2	7	2.048	8.48	1.82	1.13	3.83	1.68	0.56	2.19	2.00	1.15	3.91	5.82
101	5	24	4	2	7	2.048	10.20	1.83	1.15	3.84	1.71	0.53	2.35	2.04	1.14	3.90	5.82
97	17	5	5	3	9	8.192	2.47	13.96	5.52	36.03	12.99	5.16	45.70	14.65	5.35	34.02	41.86
257	17	15	5	3	9	8.192	10.29	16.28	6.58	47.02				14.45	5.30	23.53	41.86
401	17	24	5	3	9	8.192	13.08	16.38	7.49	44.20				14.61	5.36	23.53	41.86
97	17	5	3	4	9	8.192	2.75	22.56	12.89	109.90	18.84	5.89	23.85	23.63	12.89	68.73	69.87
257	17	15	3	4	9	8.192	10.84	23.86	12.53	116.30				22.15	10.70	62.82	69.87
401	17	24	3	4	9	8.192	13.32	20.97	9.67	54.85				19.17	8.14	55.62	69.87
97	17	5	2	5	9	8.192	2.14	23.18	9.03	38.97	25.14	7.87	95.38	24.44	9.00	57.97	94.76
257	17	15	2	5	9	8.192	10.49	27.11	11.47	75.04				24.79	9.17	52.78	94.76
401	17	24	2	5	9	8.192	13.16	28.22	13.02	78.44				24.17	9.44	54.93	94.76
121	21	5	2	5	9	8.192	2.23	23.08	9.04	39.13				24.29	8.91	57.34	94.76
321	21	15	2	5	9	8.192	10.28	26.30	11.58	76.18				23.95	9.49	54.08	94.76
501	21	24	2	5	9	8.192	12.84	27.15	12.35	67.63				23.92	9.39	38.89	94.76

and *theoretical* worst case analysis for the e2e-RT. For the sake of comparison, since GTS slots are allocated only to CHs, both e2e-BE and e2e-RT delays are computed as sum of per-hop delays of messages sent from child to parent in the tree, recursively up to the GW. A column is for showing the packet loss ratio (for BE traffic only: these values account for the number of packets whose sending failed after three retransmissions). On the contrary, due to our design and setup choices (i.e., to assign GTS slots to CH children only), RT traffic experienced no packet losses.

Although experimental results are available for scenarios with up to 101 nodes, i.e., the maximum dimension of a single WSN Patch in DEMMON1 (Fig. 10(a)), overall the following conclusions can be drawn: (i) the statistics of the e2e-BE delay match the experimental ones; (ii) the analytical tool for worst case dimensioning gives an upper bound of the maximum e2e-RT delay; (iii) as expected, while the statistics of e2e-RT delay are not influenced by the clusters' size ( $\Sigma$ ), for e2e-BE delay the impact of a more crowded network becomes not negligible; (iv) by looking at the scenarios with  $\Gamma = 17$  and by averaging among the 5 values of  $\Sigma$ , a topology with a wider ( $R_m = 5, L_m = 3$ ) rather than a deeper ( $R_m = 2, L_m = 5$ ) tree shows gains in the e2e-BE and e2e-RT delays of almost 68.2% and 66.2%, respectively. Additionally, the difference in terms of packet loss for the same scenarios is negligible. Finally, these results highlight that the EMMON network architecture scales well with the number of nodes in a WSN Patch. This is one of the most important aspects of EMMON, since it shows that our hierarchical architecture can scale easily to thousands of nodes by dividing the network into patches.

### 6.3 Network Coverage Simulation Analysis

Due to the limited area of our deployment for DEMMON1, we did not actually apply the network planning tool as described in Section 3. However, to further assess the scalability of the proposed architecture, a set of experiments have been conducted to find the minimum number of nodes needed to cover a field of size up to  $49 \text{ km}^2$ .

Each WSN Patch fits into a square cell of length  $L_{SBD} =$

**Table 3: Coverage Simulation Results - SBD**

Field Size [ $\text{km}^2$ ]	WSN Patches	SNs to cover $\geq 60\%$	SNs to cover $\geq 80\%$
1	25	600	1250
9	225	5000	10000
49	1225	30000	50000

200 m (SBD scheme) or an equilateral triangle tile of length  $L_{TBD} = 200$  m (TBD scheme), with the GWs placed in the middle, and  $C = 12$  CHs evenly spread along two rings around the GW, whose radius are  $R_1 = 40$  m and  $R_2 = 80$  m, respectively (c.f. Fig. 3). Then, a variable number of SNs are randomly distributed in the field and every node (SN and CH) associates to a parent node following the IEEE 802.15.4 association mechanism and a simple metric based on the evaluation of the received signal strength (RSSI). The RSSI measurements are simulated by relying on the inherent radio propagation and IEEE 802.15.4 interference models offered by the simulator [25] and tuned with TelosB-like physical parameters.

Given  $R = 50$  m, the communication radio range common to every node in the WSN Patch and  $R_r = 25$  m, the sensing range of each SN (Fig 5), the field coverage has been computed as described in Section 3. Results are reported in Table 3 and Table 4 for the SBD and TBD schemes, respectively. In particular, Table 3 reports on the numbers of network elements needed to cover at least 60% or 80% of the monitoring area, while in Table 4, assuming the same number of SNs as in Table 3, the focus is on the resulting coverage.

Comparing the two deployment schemes we can claim that: (i) SBD reaches an almost full coverage, while the TBD shows limited performance, but (ii) to allow neighbor patches to work in parallel, the SBD scheme requires that each patch runs on one of 9 distinct frequency channels (each patch has 8 neighbor patches); while the TBD requires only 4 channels (each patch has 3 neighbor patches). As a consequence, the best deployment approach is to be traded off

**Table 4: Coverage Simulation Results - TBD**

Field Size [ $km^2$ ]	WSN Patches	SNs	Coverage [%]
1	77	600	39.51
1	77	1250	66.37
9	527	5000	36.88
9	527	10000	60.20
49	2627	30000	36.74
49	2627	50000	52.95

with the application requirements: e.g., in a controlled environment like in a Data Center we might allow to reserve more channels for our monitoring network, while in Urban environments only the classical IEEE 802.15.4 four channels might be available. However, both tables show that the number of WSN Patches as well as the average number of SNs per patch increase quite slowly. This confirms that the proposed architecture scales well against the size of the monitored area, with relatively stringent coverage requirements.

## 7. CONCLUSIONS AND FUTURE WORK

This paper outlined our toolset to support EMMON WSN system architecture for large-scale, dense and real-time embedded monitoring. It is a complete toolset for engineering these systems, encompassing deployment planning, network dimensioning, analysis, protocol simulation and nodes testing/programming: this helps the designer to have a global picture of the whole system.

This toolset has been paramount to test the EMMON baseline system architecture through extensive simulation as well as experimental evaluation, proving its feasibility and scalability. DEMMON1, i.e., the first EMMON demonstrator, is a 300+ nodes test-bed and is, to the best of our knowledge, the largest single-site WSN test-bed in Europe.

Ongoing work about the EMMON system includes the extension of its baseline architecture with security and reliability add-ons. As a consequence, parallel work will be devoted on extending the toolset to support the design and validation of such new features. For instance, sniffer and parser tools (as mentioned in Section 5) will be extended (i) to allow the synchronization among multiple distributed sniffing points, (ii) to decode ciphered packets and, finally, (iii) to inject packets in a fully controlled way, in order to test the network's fault tolerance mechanisms.

Overall, we believe that the EMMON system architecture and the related toolset presented in this paper will foster and ease the design, test and validation of WSN applications.

## 8. ACKNOWLEDGMENTS

This work has been partially supported by (i) National Funds, through the FCT - Portuguese Foundation for Science and Technology, under grant ref. ARTEMIS-0003-2008, (ii) the ARTEMIS Joint Undertaking, under grant agreement n. 100036, (iii) a consortium of companies consisting of Critical Software S.A., Intesys Ltd, Critical Software Technologies Ltd, SESM S.c.a.r.l., Akting Ingeniaritza S.L., Centro de Estudios e Investigaciones Técnicas de Gipuzkoa, (iv) the CONET - Cooperating Objects Network of Excellence, funded by the European Commission under FP7 with grant ref. FP7-ICT-224053.

## 9. REFERENCES

- [1] Emmon - embedded monitoring. European Community's Seventh Framework Programme (FP7/2007-13). Grant agreement No. 217967. (<http://www.artemis-emmon.eu>).
- [2] Wirelessly accessible sensor populations (wasp) project. FP6-IST-2005-2.5.3 Embedded Systems, Contract Number IST-034963. Duration: Sep 2006 to Oct 2010 (<http://www.wasp-project.org>).
- [3] Open-zb: Open source toolset for ieee802.15.4 and zigbee, 2010. (<http://www.open-zb.net/index.php>).
- [4] Ekahau heatmapper - the free wi-fi coverage mapping site survey tool, 2011. (<http://www.ekahau.com>).
- [5] A. Arora, R. Ramnath, E. Ertin, P. Sinha, S. Bapat, V. Naik, V. Kulathumani, H. Zhang, H. Cao, M. Sridharan, S. Kumar, N. Seddon, C. Anderson, T. Herman, N. Trivedi, M. Nesterenko, R. Shah, S. Kulkarni, M. Aramugam, L. Wang, M. Gouda, Y. ri Choi, D. Culler, P. Dutta, C. Sharp, G. Tolle, M. Grimmer, B. Ferreira, and K. Parker. Exscal: elements of an extreme scale wireless sensor network. In *Embedded and Real-Time Computing Systems and Applications, 2005. Proceedings. 11th IEEE International Conference on*, pages 102 – 108, 2005.
- [6] R. Barr, Z. J. Haas, and R. van Renesse. Jist: an efficient approach to simulation using virtual machines: Research articles. *Softw. Pract. Exper.*, 35(6):539–576, 2005.
- [7] E. Biagioni and G. Sasaki. Wireless sensor placement for reliable and efficient data collection. In *System Sciences, 2003. Proceedings of the 36th Annual Hawaii International Conference on*, page 10 pp., jan. 2003.
- [8] M. Ceriotti, L. Mottola, G. P. Picco, A. L. Murphy, S. Guna, M. Corra, M. Pozzi, D. Zonta, and P. Zanon. Monitoring heritage buildings with wireless sensor networks: The torre aquila deployment. In *Proceedings of the 2009 International Conference on Information Processing in Sensor Networks, IPSN '09*, pages 277–288, Washington, DC, USA, 2009. IEEE Computer Society.
- [9] I. Crossbow Technology. *MICAz Mote Datasheet.*, 2009.
- [10] I. Crossbow Technology. *TelosB Mote Datasheet.*, 2009.
- [11] P. Dutta, P. Aoki, N. Kumar, A. Mainwaring, C. Myers, W. Willett, and A. Woodruff. Common sense: Participatory urban sensing using a network of handheld air quality monitors (demonstration). In *Proceedings of SenSys*, pages 349–350, Berkeley, CA, November 2009.
- [12] O. Gnawali, K.-Y. Jang, J. Paek, M. Vieira, R. Govindan, B. Greenstein, A. Joki, D. Estrin, and E. Kohler. The tenet architecture for tiered sensor networks. In *Proceedings of the 4th international conference on Embedded networked sensor systems, SenSys '06*, pages 153–166, New York, NY, USA, 2006. ACM.
- [13] J.-H. Hauer, R. Daidone, R. Severino, J. Busch, M. Tiloca, and S. Tennina. An open-source ieee 802.15.4 mac implementation for tinys 2.1, February 2011. Poster Session at 8th European Conference on

- [14] T. He, S. Krishnamurthy, L. Luo, T. Yan, L. Gu, R. Stoleru, G. Zhou, Q. Cao, P. Vicaire, J. A. Stankovic, T. F. Abdelzaher, J. Hui, B. Krogh, T. U. E. S. Krishnamurthy, L. Luo, T. Yan, L. Gu, R. Stoleru, G. Zhou, and Q. Cao. Vigilnet: An integrated sensor network system for energy-efficient surveillance. *ACM Transaction on Sensor Networks*, 2:1–38, 2006.
- [15] P. Jurcik. *Real-time Communication over Cluster-tree Wireless Sensor Networks*. Electrical engineering and information technology, control engineering and robotics, Department of Control Engineering Faculty of Electrical Engineering Czech Technical University in Prague Czech Republic, in collaboration with CISTER-ISEP Research Unit Polytechnic Institute of Porto Portugal, Prague, January 2010.
- [16] P. Jurcik, A. Koubaa, M. Alves, E. Tovar, and Z. Hanzalek. A simulation model for the iee 802.15.4 protocol: Delay/throughput evaluation of the gts mechanism. In *Modeling, Analysis, and Simulation of Computer and Telecommunication Systems, 2007. MASCOTS '07. 15th International Symposium on*, pages 109–116, oct. 2007.
- [17] P. Jurcik, A. Koubaa, R. Severino, M. Alves, and E. Tovar. Dimensioning and worst-case analysis of cluster-tree sensor networks. *ACM Transactions on Sensor Networks*, 7:14:1–14:47, September 2010.
- [18] A. Koubaa, S. Chaudhry, O. Gaddour, R. Chaari, N. Al-Elaiwi, H. Al-Soli, and H. Boujelben. Z-monitor: Monitoring and analyzing iee 802.15.4-based wireless sensor networks. In *6th IEEE LCN Workshop on Network Measurements, in conjunction with 36th IEEE Conference on Local Computer Networks (LCN 2011)*, Bonn, Germany, October 2011.
- [19] C.-J. M. Liang, J. Liu, L. Luo, A. Terzis, and F. Zhao. Racnet: a high-fidelity data center sensing network. In *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems*, SenSys '09, pages 15–28, New York, NY, USA, 2009. ACM.
- [20] C.-J. M. Liang, N. B. Priyantha, J. Liu, and A. Terzis. Surviving wi-fi interference in low power zigbee networks. In *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*, SenSys '10, pages 309–322, New York, NY, USA, 2010. ACM.
- [21] R. Monica and P. Braga. Operational requirements consolidated from end-users input and opinions. Deliverable D3.1, EMMON Project, ARTEMIS Joint Undertaking Call for proposals ARTEMIS-2008-1, Grant agreement no. 100036, February 2010.
- [22] E. F. Nakamura, A. A. F. Loureiro, and A. C. Frery. Information fusion for wireless sensor networks: Methods, models, and classifications. *ACM Comput. Surv.*, 39, September 2007.
- [23] M. Presser, A. Gluhak, D. Babb, L. Herault, and R. Tafazolli. e-sense - capturing ambient intelligence for mobile communications through wireless sensor networks. In <http://www.ist-esense.org/index.php?id=18>, 2006.
- [24] B. Raman and K. Chebrolu. Sensor networks: a critique of "sensor networks" from a systems perspective. *SIGCOMM Comput. Commun. Rev.*, 38:75–78, July 2008.
- [25] P. Scheuermann. Sidnet-swans – a simulator and integrated development platform for sensor networks applications, 2010. (<http://users.eecs.northwestern.edu/ocg474/SIDnet.html>).
- [26] E. Schoch, M. Feiri, F. Kargl, and M. Weber. Simulation of ad hoc networks: ns-2 compared to jist/swans. In *Simutools '08: Proceedings of the 1st international conference on Simulation tools and techniques for communications, networks and systems & workshops*, pages 1–8, ICST, Brussels, Belgium, Belgium, 2008. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).
- [27] C. P. Singh, O. P. Vyas, and M. K. Tiwari. A Survey of Simulation in Sensor Networks. In *Proc. of the 4th International Conf. on Computational Intelligence for Modelling Control and Automation (CIMCA)*, pages 867–872, Dec. 2008.
- [28] S. Tennina, M. Alves, P. G. Sousa, M. Santos, P. Braga, M. Bourouche, G. Carrozza, R. Monica, F. Pacheco, R. Gomes, A. Garg, and R. Severino. Evaluation report: evaluation of possible solutions, concepts for new communication methods. Deliverable D4.2, EMMON Project, ARTEMIS Joint Undertaking Call for proposals ARTEMIS-2008-1, Grant agreement no. 100036, 2010.
- [29] E. Weingartner, H. vom Lehn, and K. Wehrle. A performance comparison of recent network simulators. In *Communications, 2009. ICC '09. IEEE International Conference on*, pages 1–5, 14–18 2009.
- [30] K. Xu, H. Hassanein, G. Takahara, and Q. Wang. Relay node deployment strategies in heterogeneous wireless sensor networks: single-hop communication case. In *Global Telecommunications Conference, 2005. GLOBECOM '05. IEEE*, volume 1, page 5 pp., nov.-2 dec. 2005.
- [31] K. Xu, G. Takahara, and H. Hassanein. On the robustness of grid-based deployment in wireless sensor networks. In *Proceedings of the 2006 international conference on Wireless communications and mobile computing, IWCMC '06*, pages 1183–1188, New York, NY, USA, 2006. ACM.
- [32] K. Xu, Q. Wang, H. Hassanein, and G. Takahara. Optimal wireless sensor networks (wsns) deployment: minimum cost with lifetime constraint. In *Wireless And Mobile Computing, Networking And Communications, 2005. (WiMob'2005), IEEE International Conference on*, volume 3, pages 454–461 Vol. 3, aug. 2005.