EMMON - EMbedded MONitoring

Manuel Santos*, Stefano Tennina*, Mario Alves*, Mélanie Bouroche§, Vinny Cahill§, Délio Almeida†,
Gabriella Carrozza‡, Alex Hillǂ, Georgios Chasapis≠.
*CISTER Research Center, Polytechnic Institute of Porto (ISEP/IPP), Porto, Portugal
§Distributed Systems Group, Department of Computer Science, Trinity College Dublin, Ireland
†Critical Software, Coimbra, Portugal
‡SESM scarl, Giugliano in Campania, Naples, Italy
ǂCritical Software Technologies, Southampton. United Kingdom
≠Aristotle University of Thessaloniki, Greece
Email: {mjpsa,sota,mjf}@isep.ipp.pt, dalmeida@criticalsoftware.com, chasapis@eng.auth.gr,
{Melanie.Bouroche,Vinny.Cahill}@scss.tcd.ie, gcarrozza@sesm.it, AHill@critical-software.co.uk

Abstract — **Despite the steady increase in experimental deployments, most of research work on WSNs has focused only on communication protocols and algorithms, with a clear lack of effective, feasible and usable system architectures, integrated in a modular platform able to address both functional and non–functional requirements. In this paper, we outline EMMON [1], a full WSN-based system architecture for large–scale, dense and real–time embedded monitoring [3] applications. EMMON provides a hierarchical communication architecture together with integrated middleware and command and control software. Then, EM-Set, the EMMON engineering toolset will be presented. EM-Set includes a network deployment planning, worst–case analysis and dimensioning, protocol simulation and automatic remote programming and hardware testing tools. This toolset was crucial for the development of EMMON which was designed to use standard commercially available technologies, while maintaining as much flexibility as possible to meet specific applications requirements. Finally, the EMMON architecture has been validated through extensive simulation and experimental evaluation, including a 300+ nodes testbed.**

1. Introduction

Wireless Sensor Networks (WSNs) are universally recognized as an ideal solution in several application domains. Its potential offers a different and attractive vision of how this technology may positively influence our life, from the simplest daily activities to the most complex monitoring system. Despite the growing interest from industry and academia communities in the last decades, most of research work on WSNs has focused on protocols and algorithms, trying to address the most important aspects of a WSN as power management and network integrity (e.g., medium access control, routing, data aggregation) and their local optimisation, while only a few papers report on real(istic) applications [2] and even less on network deployments in real world scenarios.

WSN technology is still extremely expensive, which is impractical for large scale deployments and perhaps, as a consequence, there is a lack of complete and ready–to–use system architectures. Being the cost probably the major constraint, WSN technology is still unreliable, mostly because of low–cost and low–power radios operating in highly–crowded ISM bands.

EMMON [1] is an ARTEMIS industry–driven project targeting large–scale and dense real–time monitoring aiming to support applications from a wide range of domains [3], such as energy–efficiency in data centres [4] and infrastructures monitoring [5] (e.g., bridges, tunnels or the power grid). It aims to develop an integrated framework of technologies, network planning and deployment tools to support applications from with the goal of providing them with Quality–of–Service (QoS) requirements in an integrated fashion, i.e., considering scalability, timeliness, reliability and energy–efficiency.

Several relevant work on WSN systems supported by working prototypes in the scope of research projects [9]–[11] have been outlined in the literature (e.g. [4], [6]–[8]) as presented in Section 2.1. However, to the best of our knowledge, none of them fulfils all requirements for large–scale and dense real–time monitoring [11]. In this context, the EMMON system architecture advances the state of the art by combining and optimizing the following aspects:

- It builds upon user/application requirements [3], in terms of sensor number/granularity, sampling frequency and end–to–end delay, and also upon a thorough analysis of the problems to address and of previous work ([12] and [16]).
- It uses the baseline IEEE 802.15.4 [13], [14] and ZigBee [15] protocol stacks, widely–known standards, designed and implemented in synergy with the TinyOS 15.4 and ZigBee Working Groups and backed up by expertise available in the consortium.
- Maintaining as much flexibility as possible, it is built upon the most widely–used standard and commercially available COTS technologies for WSN (i.e., IEEE 802.15.4 and ZigBee), well–known technologies which increase the users' confidence and reduce the development time, granting obvious benefits for system designers and end–users.
- It enhance IEEE 802.15.4 and ZigBee with very important add–ons, such as dynamically adaptable duty–cycling, efficient cluster scheduling, traffic differentiation and downstream geographical routing (Section 4).

- It envisages of fulfilling Quality–of–Service (QoS) properties in an integrated fashion, considering scalability, timeliness (including real–time support), reliability/robustness and energy–efficiency.
- All system components are encompassed, from a Command and Control (C&C) user interface, to communication network architecture, middleware and hardware (based on COTS) platforms (Sections 4–6).
- The WSN architecture is supported by 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 (outlined in Section 7).
- The baseline architecture has been tested and validated by extensive simulation and experimental evaluation, including a 300+ node test–bed [1], which is the largest single–site WSN test–bed in Europe to date (Section 8).

This paper outlines the most relevant aspects of the EMMON system architecture, and is structured as follow. Section 2 illustrates some of the most relevant prior work related to EMMON, which served as the starting point to design the EMMON system architecture, of which an overview is provided in Section 3. Sections 4, 5 and 6 focus on the communication network architecture, middleware and C&C system. Section 7 provides an overview of the toolset that we devised and used for network planning and validation, encompassing analytical and simulation models, nodes programming and experimental data gathering. Finally, Section 8 illustrates the results of a first instantiation of EMMON (DEMMON1[1]) on a physical test–bed and assesses them against simulation and analytical models. Concluding remarks are given in Section 9, as well as an outline of the on–going work related to the instantiation of the EMMON system architecture into several application scenarios.

2 Design Approach

The WSN wide applicability fostered an unprecedented interest on all technological issues pertinent to WSNs. Hence, plenty of solutions in the literature are available encompassing all aspects of WSN–based systems, ranging from networking protocols to algorithms design and their optimisation. A thorough analysis was carried out in [12] and here briefly summarized, which was focused on WSN systems and applications involving real–world deployments [4], [6]–[10] in order to infer the best practices that could be reused to design a complete WSN system architecture (EMMON).

2.1 Outline of Some Relevant Previous Work

ExScal [6] fielded a 1000+ node WSN with an ad–hoc backbone network of 200+ 802.11–equipped devices, in a 1.3 km by 300 m remote area, for intrusion detection. This project organized the biggest WSN deployment to date and although it supports only a single application, its multi–tier network architecture is relevant to EMMON. However, the application targeted is quite different and a planned and regular topology makes the solutions adopted too specific.

VigilNet [7] was one of the major efforts in the community to build an integrated WSN system for surveillance. Its goal was to develop an operational self–organized WSN to provide surveillance with a sentry–based power management scheme, in order to achieve a minimum 3–6 month lifetime with current hardware. Although not directly related to EMMON scenarios, the energy–aware design methodology for large scale networks used has actually inspired part of our design.

Middleware for WSN and how to program them is often cited as the weakest link to rapid development and deployment of WSNs [19]. Tenet [8] investigates WSN application development simplification and software reuse. The proposed architecture is tiered, consisting of motes in the lower tier and relatively unconstrained platform nodes in the upper tier. Tenet supports only 2 tiers and this limits its scalability, as it assumes that no processing is performed at the lower tier. EMMON extends this view to multi–tier and supports processing at each tier.

RACNet [4] aims at using WSN for improving energy–efficiency in data centres with a working prototype system of almost 700 nodes. The most interesting aspect of RACNet is that it proposes a solution to maintain robust data collection trees rooted at the network's gateways. It builds upon the IEEE 802.15.4 protocol and includes an analysis of its coexistence with other technologies, such as WiFi, sharing the same band. EMMON opts for a similar approach, but instead of implementing token–based communication among the nodes, it allows for a more structured network coordination of clusters of nodes, focusing on guaranteeing a given level of QoS.

e–SENSE [9] provided heterogeneous WSN solutions to enable context capture for ambient intelligence. Three classes of applications were investigated: (a) body sensor network applications, (b) WSNs applications with and (c) without localization. The network architecture comprises various possible instantiations of mesh WSNs connected via gateways to a core network, e.g., a cellular network or a conventional wired backbone network. While three different instantiations were presented, this project does not provide a fully–implemented architecture and does not address scalability, as EMMON does.

The WASP project [10] aimed at developing a generic and portable programming model, moving from the evaluation of existing communication and security protocols, and operating systems. Beyond some differences on technical aspects (i.e., WASP uses a beacon–less MAC protocol), similarly to EMMON, the overall goal of this project was to make WSNs really usable. However, WASP's approach is to build on a set of proprietary HW/SW solutions, while EMMON strategically leverages standard and COTS technologies.

2.2 Design Guidelines

As already stated, a comprehensive analysis of existing technologies and related work is available in [12]. From such an analysis of the most important projects found in literature, we have inferred a set of lessons, i.e., best practices to develop applications for medium to large scale WSNs. From these and from our own past experience, such design guidelines to be

[1] The acronym stands for EMMON Demonstrator, phase 1.

applied to the EMMON architecture are summarized as follows: **(1)** ***Keep it simple***: simple solutions are easier to handle and debug. The interaction with the end–users helps identifying the appropriate requirements, often leading to a reduction of the complexity. **(2)** ***Modular design***: as good practice, proceeding by steps in a modular design approach is of paramount importance. A multi–cycle design process tackling the various phases of the overall process, allow verifying their correctness evaluated through analytical and simulation models, as well as experimental validation. **(3)** ***Embed tests in the design cycles***: extensive tests using a test-bed should be included in the design refinement cycles (test–it–fix–it). Many properties and problems appear only in real–world deployments [20]; therefore, it is paramount to deploy the test–bed in an environment that exhibits similar conditions as the final deployment, as well as tuning consistent simulation models. **(4)** ***Interoperability matters***: with an example, the best Medium Access Control (MAC) protocol may not fit the requirements of the best routing protocol; therefore, assessing the interoperability between technologies is important to evaluate the adequacy of each of them and in the whole system frame. **(5)** ***Technical maturity***: in engineering projects, choosing mature and standard technologies, that have been implemented and preferably extensively used by many people, is the key for the success. **(6)** ***QoS provision***: to achieve predictable resource guarantees, it is mandatory to rely on network models, such as cluster–tree, rather than mesh–like. These network models rely on (i) the use of contention–free MAC protocols (e.g., TDMA or token passing), (ii) tree–routing protocols and (iii) the possibility to reserve end–to–end resources. These are key points allowing structured and deterministic models to outperform mesh–like topologies.

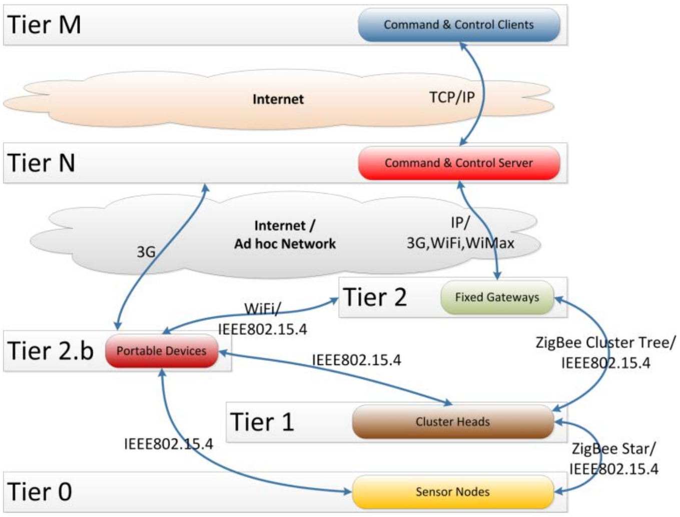


Figure 1 – EMMON High Level Multi–Tiered Hierarchical System Architecture.

3 EMMON System Architecture

A paramount characteristic of architectures for a WSN system aiming at achieving large scale is the scalability. But, what “scalability” means? Depending on characteristics such as the application, the environment or the users, the scale of a WSN system may dynamically change over time. The term “scale” applies to the number (fewer or more nodes in the overall system), spatial density (fewer or more nodes in a restricted region), or geographical region covered (smaller or wider, 2D or 3D). The ability of a WSN system to easily/transparently adapt itself with no or negligible degradation of overall system performance[2] to these dynamic changes in scale is named “scalability” [16]. Then, the main goal of EMMON is to provide an architecture for WSN systems, which is both scalable and able to fulfil QoS requirements.

By applying the best practices described in Section 2, building on the alternatives identified in [12] and to cope with scalability issues while addressing QoS requirements, our approach is to “*divide et impera*”, i.e., to adopt a hierarchical, multi–tier network architecture as sketched in Figure 1[3]. Furthermore, following from extensive consultation with experts from a wide number of fields [3], EMMON adopts a fully geographical approach: users specify the area from which they want data, as opposed to the nodes that should be queried. Its main characteristics are summarized in the following and detailed in Sections 4–6.

- The synchronized version of IEEE 802.15.4 is used at the lowest tiers. By dividing the time into active and sleep periods, this MAC helps to achieve the goals of timeliness, time synchronization and lifetime. Nodes are synchronously active or sleeping, with a dynamically adaptable duty–cycle. 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.
- WSN nodes are organized into a ZigBee–based Cluster Tree network model [21], rooted at a gateway playing the role of the sink. A cluster–tree is a hierarchical architecture per–se. However, to avoid collisions between clusters, while meeting end–to–end deadlines of time–bounded data flows, clusters’ active portions are scheduled in a non–overlapping fashion using the Time Division Cluster Scheduling (TDCS) [21].
- We adopt tree–routing for upstream traffic, which has a negligible memory footprint and processing delay since no routing tables are needed. We also support efficient geographical–based routing of queries for the downward flow, for

[2] End–to–end delay, throughput, security, reliability and lifetime.

[3] It includes an optional element, i.e., the PDA belonging to an intermediate Tier2.b and playing a special role of mobile gateway and diagnosis element. However, details about this component are out of scope of this paper.

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 (Section 5), rather than any explicit request for raw readings from specific sensor nodes.

- Beaconing inherently enables the support for time synchronization at the Data Link and Network layers. This enables accurate time stamping of sensor data (required by many applications), energy–efficiency through duty–cycling, cluster scheduling techniques and contention–free MAC.
- Data aggregation, sensor and data fusion mechanisms [22] are implemented at all levels of the architecture: (1) at the sensor nodes (SNs), by aggregating multiple readings taken over time (temporal aggregation), before sending these data to the parent; (2) at the cluster heads (CHs), by aggregating multiple readings coming from different sensors or children CHs (spatial aggregation), before forwarding the report to the parent; (3) at the gateway (GW), where sensor fusion, i.e., inferring useful information through, e.g., model fitting, is potentially done by considering multiple reports coming from the CHs, before sending them to the C&C. (4) at the C&C, where 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 [3]).
- A novel EMMON–specific middleware (Section 5) 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 (Figure 1). 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.
- The EMMON C&C subsystem is the interface to the end–users (Section 6). 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.

4 Communication Protocol Architecture

The extensive analysis conducted in [12] and briefly summarized in Section 2 constituted the starting point to derive appropriate network architecture, to achieve efficiency in large scale and dense WSNs for EMMON's purposes. In particular, a set of alternative networking stacks/technologies were identified, which have the following common features: (1) a multitier architecture, and (2) IEEE 802.15.4 for short–range communications.

Adopting the IEEE 802.15.4 standard was a natural choice for EMMON. However, the use of a multi–tiered architecture raises a number of challenges, although considered by far the best network architecture for the purpose, thanks to the flexibility it offers. In particular, two issues must be tackled: (i) how many tiers and how many communication technologies are to be used, and (ii) what kind of nodes are the most appropriate for each tier (e.g., in terms of computing power and power supply type).

4.1 Design Choices

In order to achieve scalability and a given degree of QoS while maintaining a low level of complexity in the network, several assumptions were made. At the higher tiers of the system, IP is used as the base networking protocol and our architecture supports the case where one or more gateways, equipped with e.g., WiFi or 3G for long range communications constitute a backbone–based (ad hoc) network as well as the possibility for them to be in "direct" communication with a remote C&C server over the Internet or any kind of public network (Figure 1). At the lowest tiers we assume a clustered architecture, since clusters [23]: (i) help localize routes and reduce the size of the routing table, (ii) conserve bandwidth and prolong battery life through duty cycling and (iii) result in a reduction in coverage redundancy, medium access collisions, transmission range required and/or number of hops required to reach the sink. While the random node deployment paradigm is appealing for large–scale WSNs due to its inherent low deployment costs, if nodes are randomly scattered some of them might be unreachable or have to use a very high transmission power to maintain network connectivity, typically resulting in faster battery exhaustion. Therefore, EMMON assumes some control over node deployment and thus the ability to ensure that nodes are relatively evenly spread; in particular cluster heads are placed in order to maximize network connectivity. For that purpose we have devised system planning tools like the one described in Section 7. The system also relies on nodes being position–aware: either they know their own position pre–run–time (e.g., it is a parameter configured at deployment time, as in [7]), or they are able to estimate it using some positioning service [24] provided by the middleware. Finally, given the type of applications targeted by EMMON and since no end–user typically expressed requirements for peer–to–peer communication [3], the EMMON architecture does not support horizontal data flows. Therefore, EMMON only supports communication from nodes to the C&C (i.e., upward flow), to send measurements reports and alarms notifications, and from the C&C to nodes (i.e., downward flow), for disseminating user–defined operations, network management commands or to reconfigure/reprogram at run–time group of nodes. Downstream control (i.e., for actuation) or network management traffic may also be supported, if required.

4.2 Multi–tiered Architecture

Given the above design choices, the resulting EMMON architecture is as sketched in Figure 1. Tier–0 consists of simple wireless sensor nodes, performing sensing tasks and delivering data to the devices at the upper tier in the hierarchy using the IEEE 802.15.4 protocol. It is expected they are cheap enough to be deployed in large quantities, therefore we assume they have very limited computational, memory and energy capabilities.

Multiple SNs are grouped to form a WSN Cluster at Tier–1 in a star topology, where a Cluster Head is responsible for cluster management. Cluster heads may be slightly more powerful than ordinary sensor nodes in terms of computational capabilities, and might have better energy reserves or be powered by an auxiliary energy source.

Multiple CHs are grouped to form a WSN Patch at Tier–2, where a fixed gateway is present. GWs should have the highest computational capabilities in the WSN and play the role of sinks/roots for a WSN Patch. GWs are assumed to be equipped with a secondary transceiver, e.g. a WiFi radio, and to have a direct connection(e.g., ad–hoc multi–hop) to the C&C Server at Tier–N, when the C&C is physically close enough to the sensor field, or an indirect connection (e.g., through the Internet), when the C&C is remote.

A WSN Patch adopts a Cluster–Tree model, with a GW as root and the SNs as leaves. As discussed in Section 3, the synchronous version of the IEEE 802.15.4 protocol was chosen to easily support time synchronization, duty–cycling and guaranteed bandwidth.

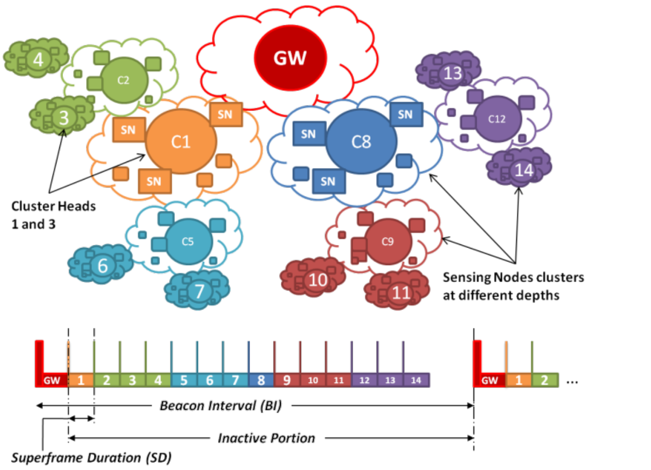


Figure 2 – TDCS: the active portion of a cluster is scheduled during the inactive portion of all others.

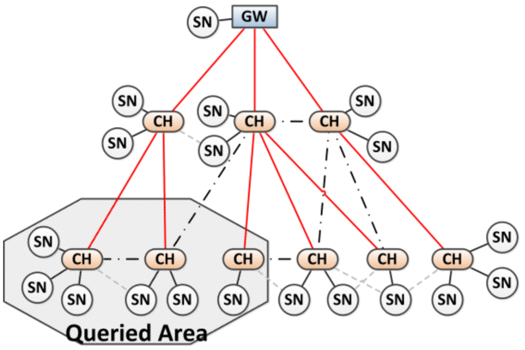


Figure 3 – Cluster Tree of a WSN Patch. Continuous line indicates Parent–Child relations; dashed line stresses that deployment planning should allow for a child to have alternative parents.

Beacons are messages sent by every local coordinator in the 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 [21]. However, to preserve the coordination and avoid intra–cluster collisions, the TDCS algorithm [21] 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 Figure 2. This design choice leads to a given upper bound on the allowed number of clusters (Γ[4]) and related duty cycles (DC).

4.3 Networking

As already stated, the EMMON architecture comprises different types of devices, ranging from low power sensor nodes to more powerful fixed gateways and C&C stations (server and clients). Then, to enable networking among these heterogeneous devices, since IP is used in the higher tiers, the most appropriate addressing scheme for sensor nodes and cluster heads must be chosen.

EMMON adopts the simple ZigBee–based addressing mechanism. However, as nodes must be reached from the C&C, which uses only IP, address translation must be supported. Our proposal differentiates between the multipoint–to–point upward flow and the point–to–multipoint downward flow. Assuming that the root (GW) for each patch is given a default local address (0x0000), upward flow reduces to the simplest converge cast routing over the tree. On the contrary, since nodes know their geographical position, for the down–stream flow a geo–routing mechanism is used to disseminate queries, commands or (re–)configurations through the entire network.

Since the deployment of CHs is assumed to be controlled, every node is pre–programmed with its role: either GW, CH or SN. Thus, at network setup all nodes other than the GWs are scanning the medium, waiting to capture a beacon, while each GW starts by emitting its beacons using a predefined IEEE 802.15.4 channel. As soon as some CHs receive a GW’s beacons, they start the association process with the GW, in accordance with the IEEE 802.15.4 protocol and acting as normal nodes. Once associated with the parent, they start a negotiation procedure to obtain an appropriate window, in which they can transmit their own beacons in a non–overlapping fashion with other CHs [21]. Hence, this mechanism iteratively enables all other nodes (SNs and other CHs[5]) to participate in the network upon a successful association phase, forming a Cluster–Tree topology, as shown in Figure 2 and Figure 3. This enables efficient fault tolerance mechanisms, such as pro–active re–association mechanisms, based on a periodic link quality estimation [25]. To allow for the above–mentioned geo–routing mechanism, every time a node (either SNs or CHs) associates with a parent (either a CH or the GW), it communicates its own position, so that the parent can compute its Served Area (SA), i.e., the area encompassing its child sensing devices.

[4] The total number of clusters in a WSN Patch must include also the GW, as a special cluster head.

[5] The CHs not in direct communication with the GW or those for which the association process failed for any reason.

5 Middleware

The EMMON architecture provides a middleware layer (EMW) to facilitate the development of our target class of applications. Due to the very–constrained nature of SNs, the choice and implementation of the services that it provides must be highly optimized. This section first presents EMW's high–level API.

The middleware API was careful designed after consultation with environmental monitoring experts from different fields [3] to capture the functionalities that are required by them, thereby enabling the middleware to optimize its internal mechanisms' non–functional properties. In particular, EMW provides a fully geographical data service, i.e., users specify the area from which they want data, as opposed to the nodes that should be queried. Users can make use of three types of operations: queries, reports, and alarms which provide data respectively once–off, periodically, and when a user specified condition is met. As a consequence, EMMON supports both periodic reporting and event–driven applications.

The middleware spans all the tiers of the architecture defined in Section 3. The functionalities differ in every tier, with many of them being implemented on several tiers, as can be seen in the overall architecture (see Figure 4). Functionality placement is a challenging design decision due to two conflicting principles. On one hand, since higher tiers are composed of less resource–constrained nodes, most computation should be performed at this level. On the other hand, placing intelligence as low as possible in the network architecture decreases the traffic volume, allows faster reaction to failures and enables their containment, hence decreasing overall complexity all characteristics that enhance scalability. The consequences of both these principles need to be weighted carefully. For example, data aggregation is performed at every tier (and even within tiers, for example at every hop in the cluster head tree), because of its potential to significantly reduce traffic volumes.

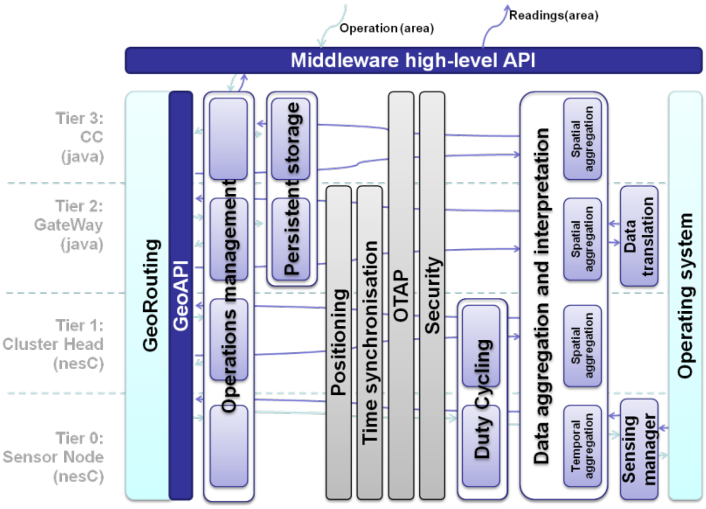
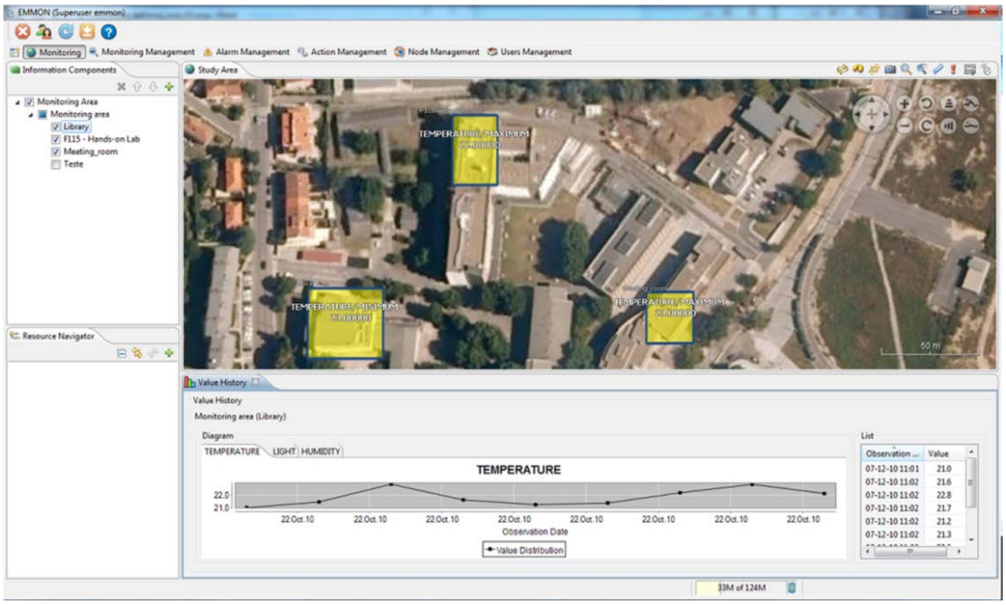


Figure 4 - EMW architecture: the lightly colored boxes are components from other software layers. White boxes were implemented for DEMMON1.

6 Command and Control (C&C)

The EMMON C&C is the most visible part of the system. It aims at allowing monitoring of a (unlimited) number of sensors and provides all the functionalities available in the WSN to end–users. For that, it is composed of two main components: the Server and the Clients.



- ☑ Monitoring Area
  - Monitoring area
    - ☑ Library
    - ☑ F115 - Hands-on Lab
    - ☑ Meeting_room
    - □ Teste

Value History

Monitoring area (Library)

List

Observation ...	Value
07-12-10 11:01	21.0
07-12-10 11:02	21.6
07-12-10 11:02	22.8
07-12-10 11:02	21.7
07-12-10 11:02	21.2
07-12-10 11:02	21.3

Figure 5 – C&C Graphical User Interface: it allows to define the monitoring areas for querying real–time sensor measurements and see the historical data

The C&C Server is responsible for interacting with the WSN, storing the data received from it and making available to the C&C Clients all the data and functionalities of the WSN. It is a central repository which contains a module responsible for receiving the data from the sensors, storing it in a database and notifying the clients of new values. It also includes a middleware component that implements the high level middleware API that is used by the C&C to interact with the WSN.

C&C Clients are the end points of the system, showing (visually) the WSN data and providing functionality to interact with the WSN. Unlike traditional software, the EMMON C&C Clients do not interact with each sensor individually but with monitoring objects (e.g., a room), which can group several sensors. This approach allows the possibility to manage WSNs with a huge number of SNs, which would be intractable with traditional approaches.

7 EM–Set – EMMON toolset, Network Planning and Analysis Toolset

This section outlines the toolset that was designed for network dimensioning and analysis, nodes testing and programming, etc. Figure 6 provides an overall perspective of the EMMON toolset, illustrating the integration between the different components (inputs/outputs) into a single framework. A brief description for each component is in what follows.

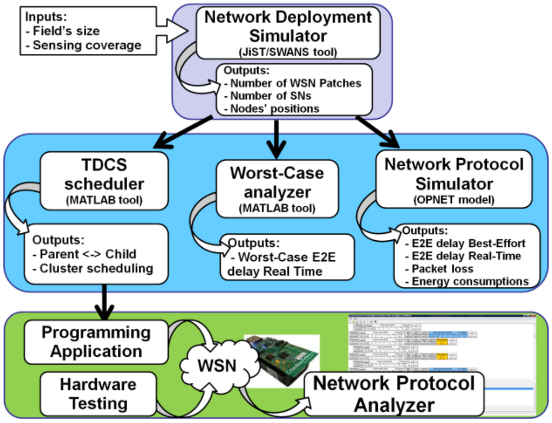


Figure 6 - EMMON toolset for network planning, dimensioning simulation analysis and test/programming.

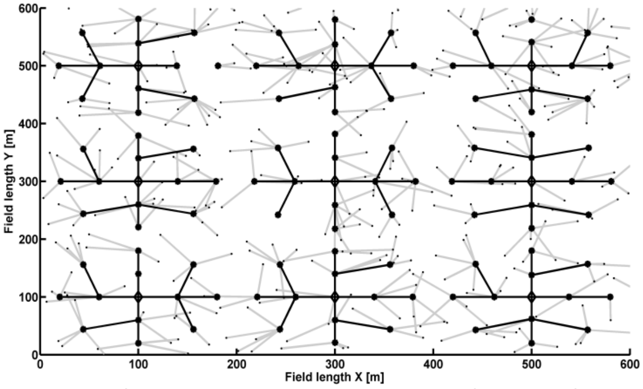
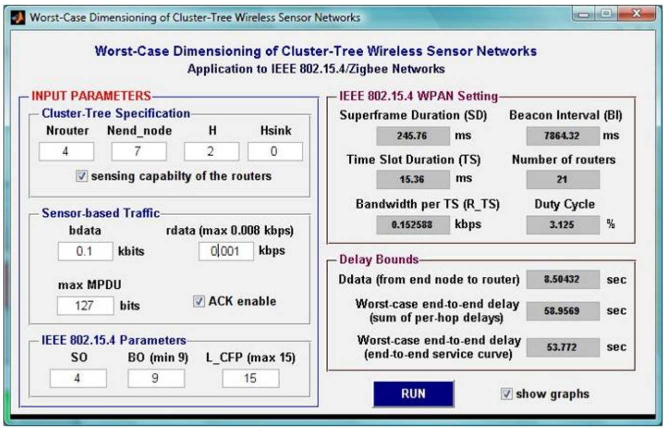


Figure 7 - Example of WSN Patches to cover at least 80% of a 49 km2 field. Diamond markers are GWs; stars are CHs, distributed along 2 hops around each GW; points are SNs, randomly deployed.

Network Deployment Simulator. Starting from a set of defined target inputs, i.e., the size of the field to monitor and a desired level of coverage, the "Network Deployment Simulation" tool outputs how many GWs, CHs and SNs are needed to meet those requirements and where to put them, assuming the Cluster–Tree network model described in Section 4. We built upon an Open Source tool named SiDNet-SWANS [26], built over JiST, a java based discrete event simulator.



Worst-Case Dimensioning of Cluster-Tree Wireless Sensor Networks

Worst-Case Dimensioning of Cluster-Tree Wireless Sensor Networks
Application to IEEE 802.15.4/Zigbee Networks

INPUT PARAMETERS

Cluster-Tree Specification

Nrouter	Nend_node	H	Hsink
4	7	2	0

☑ sensing capabilty of the routers

Sensor-based Traffic

bdata: 0.1 kbits
rdata (max 0.008 kbps): 0.001 kbps
max MPDU: 127 bits
☑ ACK enable

IEEE 802.15.4 Parameters

SO	BO (min 9)	L_CFP (max 15)
4	9	15

IEEE 802.15.4 WPAN Setting

Superframe Duration (SD): 245.76 ms
Beacon Interval (BI): 7864.32 ms
Time Slot Duration (TS): 15.36 ms
Number of routers: 21
Bandwidth per TS (R_TS): 0.152588 kbps
Duty Cycle: 3.125 %

Delay Bounds

Ddata (from end node to router): 8.50432 sec
Worst-case end-to-end delay (sum of per-hop delays): 58.9569 sec
Worst-case end-to-end delay (end-to-end service curve): 53.772 sec

Figure 8 - MATLAB tool for worst–case analysis and dimensioning

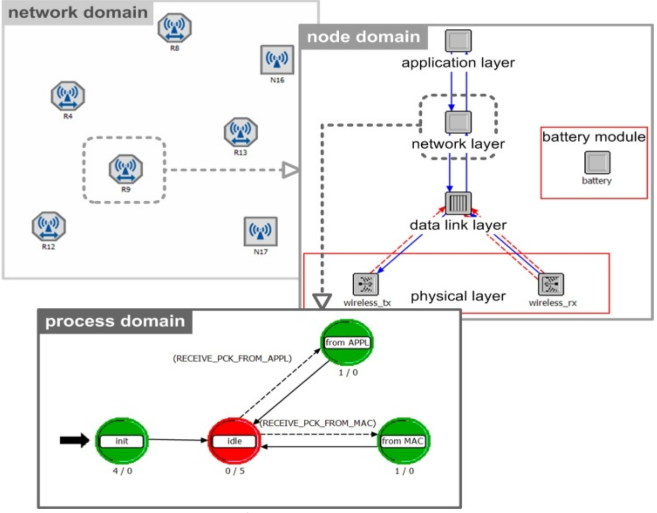


Figure 9 - The structure of the IEEE 802.15.4/Zig-Bee Opnet simulation model.

This deployment planning tool takes into account that a desired field coverage requirement must be achieved under realistic signal propagation conditions and that each resulting child should have at least two alternative candidate parents (an example is provided in Figure 7). 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 outlined next.

Worst–Case Analyzer. The Worst–Case Analyzer is a MATLAB tool (Figure 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. This tool builds on the Network Calculus mathematical model, as described in [21] and enables to find the best duty–cycle vs. delay/throughput tradeoff.

Network Protocol Simulator. The performance of the communication protocol is evaluated using a simulator built in OPNET (Figure 9) [18]. With the help of this simulator, we can infer 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.

TDCS Scheduler. This tool is built in MATLAB to compute the parent–child relation in the Cluster–Tree topology and the start time for each cluster head to schedule its active portion so that it does not overlap with the other clusters.

With the TDCS tool, a set of appropriate scripts and source files to program the nodes (currently through a USB tree) are generated. We also designed an application to automate the testing of the hardware[6] (USB cabling/hubs and TelosB nodes).

Finally, with the help of sniffer devices like [27] 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 end–to–end delays, as showed in next section.

8 Validation

This section illustrates the results of the simulation and experimental campaigns to validate the EMMON architecture and to investigate its performance and scalability limits. With respect to the EMMON network architecture, the network coverage analysis to plan the network deployment has to be addressed at the WSN Patch level in order to define: (i) the number of patches needed to cover the entire area under monitoring and (ii) how to geographically organize the patches.

Currently our Network Deployment Simulation tool uses a simple square–based deployment. While other possibilities are available, e.g., hexagonal, this solution considers the field as a grid, where each WSN Patch fits into a cell. We will rely on an example to explain how it works. Each WSN Patch occupies a square cell of length L=200 m, with the GWs placed in the center, and C=12 CHs evenly spread along two rings around the GW, whose radius are R1=40 m and R2=80 m, respectively (c.f. Figure 7). With these basic assumptions we aim at evaluating how many WSN Patches and SNs are needed to cover different field sizes. For this purpose, 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)[7]. The RSSI measurements are simulated by relying on the inherent radio propagation and IEEE 802.15.4 interference models using the simulator [26] tuned with TelosB–like physical parameters.

Given R=50 m, the communication radio range common to every node in the WSN Patch and X=25 m, the sensor range of each SN (i.e., a SN can capture environmental parameters such as temperature and humidity in a surrounding circular area of radius X), a region in the field is “covered” if there is at least a SN less than X meters away, from which an active (multi–hop) path to at least one gate-way exists. The field coverage is then defined as the ratio between the total area of the covered regions and the whole field area. Experiments have been conducted to evaluate the minimum number of nodes needed to cover field sizes up to 49 km^2. Results are reported in Table 1, which shows the dimensioning of the network with respect to the field size, for coverage values equal to 60% and 80%. 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 with respect to the size of the monitored area, with relatively stringent coverage requirements.

Field Size [Km^2]	WSN Patches	SNs to cover ≥ 60%	SNs to cover ≥ 80%
1	25	600	1250
9	225	5000	10000
49	1225	30000	50000

Table 1 - Network dimensioning for a given coverage.

Parameter	Range	Meaning
R_m	{2;3;4;5}	Number of children CH per parent
Γ	{5;17;21}	Number of Clusters
∑	{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 ration

Table 2 - WSN Patch Validation Setup

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 (Section 7), several scenarios were identified as all the combinations of the parameters in Table 3. This setup generated different network topologies with maximum depth[8] (L_m) ranging from 2 to 5 and total number of nodes in one WSN Patch ranging from 25 to 501. In the simulated scenarios, at their application layer the nodes generate and send to their parent 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 in-tended to simulate periodic sensor reports, while RT traffic accounts for 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 (Section 5).

The value of the Superframe Order (SO) has been fixed to 4, which means an active portion of 960 symbols, while the Beacon Order (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 (Γ). We assume that every cluster in a WSN Patch has the same value of the couple (BO, SO), as in Figure 2. Finally, we assume that the size of the collision domain is as large as the network size, i.e., nodes can interfere with each other. As a consequence, the TDCS scheduler gives a conservative solution, where clusters’ active portions do not overlap[9]. As a consequence, in our setup (Table 2), BO ranges between 7 and 9, giving beacon interval (BI) values of ~ 2 and ~ 8 s, respectively, and duty cycles of 12.5% and 3.125%, respectively.

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

[6] This testing tool was fundamental to early identify: (i) 19 over 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.

[7] In the near future we plan to integrate a more complete and reliable evaluation metric, like [27].

[8] Maximum number of hops between a SN and the GW.

[9] 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.

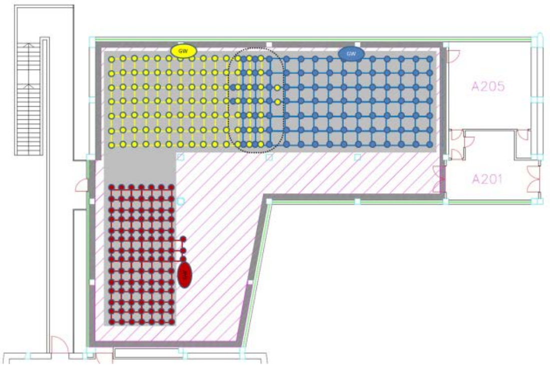


Figure 10 - DEMMON1 Deployment – 300+ nodes divided into 3 WSN Patches.

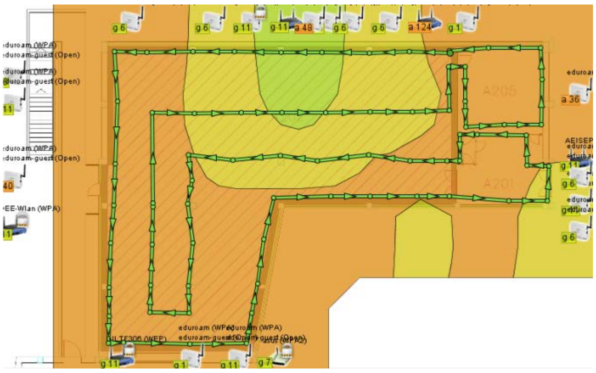


Figure 11 - DEMMON1 Deployment – ISM spectrum (namely, WLAN) interference study in the site's area

Simulation results have been compared with our DEMMON1 physical deployment (Figure 10). 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, namely ch.15, ch.25 and ch.26, chosen as they are less prone to the actual external interference. This was confirmed by a pre–deployment analysis of the interference in the deployment site (Figure 11).

Tot Nodes	Γ (clusters)	Σ (SNs per CH)	Rm	Lm	BO	BI [s]	CAP - Best Effort							CFP - Real Time			
							Pkt Loss [%]	*End-To-End Delay [s]*						*End-To-End Delay [s]*			
							SIMULATION	*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
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	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
401	17	24	2	5	9	8.192	13.16	28.22	13.02	78.44				24.17	9.44	54.93	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

Table 3 - Excerpt of results from the campaign defined in Table 2: simulation, worst–case and experimental results.

Sensor readings reports were defined on a C&C client and these were forwarded to the WSN Patches through the C&C server (using the middleware API). The packets to query the WSNs (downward flow) and the related reports (upward flow) followed the rules described in Section 4. In these experiments, 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 [27]) to compute the statistics for the e2e delay. Table 3 shows an excerpt of the results related to net-work 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 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 ac-count 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 loss.

Although experimental results are available for scenarios with up to 101 nodes, i.e., the maximum dimension of a single WSN Patch in DEMMON1, 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 (∑), for e2e-BE delay the impact of a more crowded network becomes not negligible; (iv) by looking at the scenarios with Γ =17 and by averaging among the 5 values of ∑, 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.

9 Conclusion

The EMMON WSN system architecture combines hardware platforms, communication protocols, middleware and C&C components together to keep as much flexibility as possible while meeting specific applications requirements. Additionally, we devised a complete toolset for engineering these systems, encompassing deployment planning, network dimensioning, analysis, protocol simulation and nodes testing/programming. We tested the EMMON baseline system architecture through extensive simulation as well as experimental evaluation, proving its feasibility and scalability. DEMMON1, the first EMMON demonstrator, is a 300+ nodes test-bed. Ongoing work includes the extension of the baseline architecture presented in this paper with reliability, security and data aggregation add-ons. We have also been instantiating this baseline architecture in several application scenarios, by adapting and fine–tuning some of its functionalities, namely for structural health monitoring, energy efficient management in data centers and environmental monitoring. WSNs are an appealing alternative to wired solutions for structural health monitoring, but struggle to fulfill some of the most demanding requirements imposed by these applications, such as low–power and low–cost yet extremely sensitive and accurate accelerometers and signal acquisition hardware, stringent time synchronization and system scalability. Recent results show that an EMMON–like architecture proved to be accurate and scalable, when compared to a reference wired system. We have also been tuning the EMMON system architecture to help to reduce the energy footprint in data centers. The EMMON architecture inherently supports the required fine–grained (large–scale and dense) monitoring of physical parameters in real–time, to enable identification, model, analysis and optimization of energy costs, as well as to enable dynamic reallocation of computing loads for achieving such energy and cost minimization. In parallel, the EMMON system is being applied to environmental monitoring (out-door/indoor), enabling e.g., the large–scale and dense real–time monitoring of metrics such as levels and distribution of Ozone, CO, CO_2 and acoustic noise in highly populated areas.

EMMON project will be concluded with a final demonstrator in February 2012. This demonstrator will be composed by a network of several hundreds of nodes distributed along three floors of SANJOTEC[10], a scientific park near Porto in Portugal.

It will includes new integrated features at the Middleware and C&C levels, including asynchronous queries, temperature maps, Over The Air Programming and remote restarts of the nodes.

Overall, we believe that the EMMON system architecture reported in this paper will foster and ease the design of WSN applications.

10 References

[1] EMMON - embedded monitoring. European Community's Seventh Framework Programme (FP7/2007-13). Grant agreement No. 217967. [Online]. Available: http://www.: http://www.artemis-emmon.eu

[2] B. Raman and K. Chebrolu, "Censor networks: a critique of "sensor networks" from a systems perspective, "SIGCOMM Comput. Commun. Rev., vol. 38, pp. 75–78, July 2008.

[3] R. Monica, P. Braga, "Operational requirements consolidated from end-users input and opinions", EMMON Project, D3.1, October 2011.

[4] C.-J. M. Liang, J. Liu, L. Luo, A. Terzis, and F. Zhao, "Racnet: a high-fidelity data center sensing network," inProceedings of the 7th ACM Conference on Embedded Networked Sensor Systems, ser. SenSys '09. New York, NY, USA: ACM, 2009, pp. 15–28.

[5] 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, ser. IPSN '09. Washington, DC, USA: IEEE Computer Society, 2009, pp. 277–288.

[6] A. Arora, R. Ramnath et al., "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, August 2005, pp. 102–108.

[7] 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, "Vigilnet: An integrated sensor network system for energy-efficient surveillance,"ACM Transaction on Sensor Networks, vol. 2, pp.1–38, 2006.

[8] 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 SenSys '06, pages 153–166, New York, NY, USA, 2006. ACM.

[9] 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.

[10] Wirelessly accessible sensor populations (wasp) project. FP6-IST-2005-2.5.3 Embedded Systems, Contract Number IST-034963.

[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 SenSys, pages 349–350, Berkeley, CA, November 2009

[12] S. Tennina, M. Alves, P. G. Sousa, M. Santos, P. Braga, M. Bouroche, G. Carrozza, R. Monica, F. Pacheco, R. Gomes, A. Garg, and R. Severino, "Evaluation report: evaluation of possible solutions, concepts for new communication methods," EMMON Project, D4.2, 2010.

[13] A. Cunha, A. Koubaa, R. Severino, and M. Alves, "Open-zb: an open-source implementation of the ieee 802.15.4/zigbee protocol stack on tinyos," in Mobile Adhoc and Sensor Systems, 2007. MASS 2007. IEEE Internatonal Conference on, 2007, pp. 1 –12.

[14] J.H. Hauer, R. Daidone, R. Severino, J. Busch, M. Tiloca, and S. Tennina. (2011, February) An open-source ieee 802.15.4 mac implementation for tinyos 2.1. Poster Session at 8th European Conference on Wireless Sensor Networks.

[15] Zigbee Specifications, ZigBee Standards Organization Std. 053 474, Rev. 17, January 2008. [Online]. Available: http://www.zigbee.org

[16] M. Alves, P. G. Sousa, M. Bouroche, P. Braga, R. Monica, G. Carrozza, S. Prabh, and R. Gomes, "Study of collected, analysed and classified problems to address in this project," EMMON Project, D4.1, 2009

[17] 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.

[18] Open-zb: Open source toolset for ieee802.15.4 and zigbee, 2010. (http://www.open-zb.net/index.php)

[19] L. Mottola and G. P. Picco. Programming wireless sensor networks: Fundamental concepts and state-of-the-art. ACM Comp. Surveys, 43(3), 2011.

[20] K. Langendoen, A. Baggio, and O. Visser. Murphy loves potatoes: experiences from a pilot sensor network deployment in precision agriculture. In IPDPS2006, page 8 pp., 2006

[21] P. Jurcik, R. Severino, A. Koubaa, M. Alves, and E. Tovar. Dimensioning and worst-case analysis of cluster-tree sensor networks. ACM Transactions on Sensor Networks, 7(2), August 2010.

[22] E. F. Nakamura, A. A. F. Loureiro, and A. C. Frery. Information fusion for wireless sensor networks: Methods, models, and classifications. ACM Computing Surveys, 39, September 2007.

[23] A. A. Abbasi and M. Younis. A survey on clustering algorithms for wireless sensor networks. Computer Communications, Special Issue on Network Coverage and Routing Schemes for Wireless Sensor Networks, 30:2626–2841, 2007.

[24] S. Tennina, M. Di Renzo, F. Graziosi, and F. Santucci. Locating Zigbee nodes using the ti's cc2431 location engine: a testbed platform and new solutions for positioning estimation of WSNs in dynamic indoor environments. In ACM MELT '08, pages 37–42, New York, NY, USA, 2008. ACM.

[25] N. Baccour, A. Koubâa, H. Youssef, M. Ben Jamâa, D. do Rosário, M. Alves, and L. Becker. F-LQE: A Fuzzy Link Quality Estimator for Wireless Sensor Networks. In (EWSN'2010). September 2010.

[26] P. Scheuermann. Sidnet-swans – a simulator and integrated development platform for sensor networks applications, 2010.

[27] Z-monitor: A monitoring tool for ieee 802.15.4 wireless personal area networks, 2010. (http://www.z-monitor.org).

[10] SANJOTEC, the new Science & Technology Park of S. João da Madeira (www.sanjotec.com), located approximately 15km from Porto, Portugal, which aims is to support technically and scientifically the local and regional enterprise community, through the diffusion of an innovation culture and the encouragement of enterprising projects of technological basis.