Ad Hoc Networks 000 (2016) 1–18

Contents lists available at ScienceDirect

Ad Hoc Networks

journal homepage: www.elsevier.com/locate/adhoc

On optimal resource allocation in virtual sensor networks

Carmen Delgado[a], José Ramón Gállego[a], María Canales[a], Jorge Ortín[b], Sonda Bousnina[c], Matteo Cesana[c,*]

[a] Aragón Institute of Engineering Research, Universidad de Zaragoza, Spain
[b] Centro Universitario de la Defensa, Zaragoza, Spain
[c] Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, Milano, Italy

ARTICLE INFO

Article history:
Received 30 October 2015
Revised 10 March 2016
Accepted 16 April 2016
Available online xxx

Keywords:
Wireless sensor networks
Virtualization
Resource allocation
Internet of things
Optimization

ABSTRACT

Sensor network virtualization is a promising paradigm to move away from highly-customized, application-specific wireless sensor network deployment by opening up to the possibility of dynamically assigning general purpose physical resources to multiple stakeholder applications. In this field, this paper introduces an optimization framework to perform the allocation of physical shared resources of wireless sensor networks to multiple requesting applications. The proposed optimization framework aims to maximize the total number of applications which can share a common physical network, while accounting for the distinguishing characteristics and limitations of the wireless sensor environment (limited storage, limited processing power, limited bandwidth, tight energy consumption requirements). Due to the complexity of the optimization problem, a heuristic algorithm is also proposed. The proposed framework is finally evaluated by simulation considering realistic parameters from actual sensor nodes and deployed applications to provide a detailed performance evaluation and to assess the gain involved in letting multiple applications share a common physical network with respect to one-application, one-network vertical design approaches.

1. Introduction

In the Internet of Things (IoT) vision, the Internet is "pushed down" to everyday objects which are equipped with sensing capabilities to gather information on the environment they are immersed in, processing/storage capabilities to locally filter and store data, and communication peripherals to deliver the collected/processed data remotely either directly, or through multi-hop paths leveraging the cooperation of other smart objects for traffic relaying. In this last case, network of smart objects, often referred to as Wireless Sensor Networks, are set up to collect and deliver data in specific areas. WSNs can be deployed in diverse scenarios and environments to support diverse application/services ranging from smart home or environmental monitoring based on scalar sensed data to more demanding applications based on multimedia sensors.

Usually, WSNs are designed and deployed in a "vertical", application-specific way, in which the hardware and network resources are customized to the specific application requirements. On one hand, such design paradigm allows to have "optimal" performance on the specific application, but, on the other hand, it precludes resources (hardware and software) reuse when other applications and services must be contemplated. In the end, this has led in the past to the proliferation of redundant WSNs deployments [1].

In this context, novel approaches are recently being investigated targeting the smart reuse of general purpose wireless sensor networks to dynamically support multiple applications and services. The key idea behind these approaches, which often go under the names of Virtual Sensor Networks (VSN) [2] or Software Defined Sensor Networks (SDSN) [3], is to decouple the physical infrastructure and resources from application ownerships which leads to more efficient resource utilization, lower cost and increased flexibility and manageability in WSN deployments [4]. Network virtualization technologies are used to abstract away "physical resources" including node processing/storage capabilities, available communication bandwidth and routing protocols, which can then be "composed" at a logical level to support usage by multiple independent users and even by multiple concurrent applications [5]. While network virtualization and software defined networks are already a reality in many communication networks [6,7], research on sensor network virtualization is still in its infancy and comprehensive

* Corresponding author. Tel.: +390223993695; fax: +390223993413.
E-mail address: matteo.cesana@polimi.it, cesana@elet.polimi.it (M. Cesana).

http://dx.doi.org/10.1016/j.adhoc.2016.04.004
1570-8705/

solutions still need to be found to cope with the specific characteristics of WSNs in terms of limited node capabilities and communication bandwidth.

In this work, we focus on the design of a virtualization engine for WSNs. Namely, we consider a general purpose WSN which can be used to support multiple applications and we propose a mathematical programming framework to optimally allocate shared physical resources to the requesting applications. In more details, the proposed framework allocates the physical resources of the general purpose WSN to multiple concurrent applications while accounting for the network- and hardware-specific constraints (processing, storage, available bandwidth, limited communication range) and the specific application requirements. Due to the high computational complexity of the resulting optimization model, a heuristic algorithm is also proposed. Numerical results are then obtained by applying the proposed framework to realistic WSN instances to assess the efficiency of the virtualization process.

The paper is organized as follows: Section 2 overviews the related work in the field of sensor network virtualization. Section 3 describes the proposed system model and the optimization problem for resource allocation in virtual sensor networks, including a complexity analysis of this problem. Section 4 explains the proposed heuristic algorithm. In Section 5, the proposed optimization model and heuristic algorithm are evaluated by simulation for a set of scalar and multimedia applications also with different types of sensor nodes. Finally, some conclusions are provided in Section 6.

2. Related work

The emergence of shared sensor networks has stimulated research efforts in the field of novel programming abstractions at the node level and management framework at the network level to support multiple applications over a shared physical infrastructure [8–11].

At the node level, architectures based on virtual machines are proposed to enable virtualization and re-programmability. As an example, Matel̀ [12], ASVM [13], Melete [14] and VMStar [15] are frameworks for building application-specific virtual machines over constrained sensor platforms.

At the network level, several virtualization management platforms have been introduced. SenSHare [16] creates multiple overlay sensor networks which are "owned" by different applications on top of a shared physical infrastructure. UMADE [17] is an integrated system for allocating and deploying applications in shared sensor networks based on the concept of Quality of Monitoring (QoM). Fok *et al.* [18] introduce middleware abstractions to represent multiple QoM requirements from multiple applications, whereas a service-oriented middleware is presented in [19] to address the challenges faced by running multiple applications onto heterogeneous WSNs. A prototype of Software Defined Wireless Sensor Network is proposed in [20] where a centralized control plane dynamically manages communication routes in the network with the goal of augmenting the energy efficiency.

Generally speaking, the aforementioned work provides "practical" building blocks to build up virtual sensor networks. Differently, we focus in this paper on the "intelligence" to properly allocate physical resources to virtual applications, which can be cast as a general resource allocation problem. Even if radio/network resource allocation is a widely debated topic in the literature, still very few works have appeared on the optimal resource allocation in the field of Virtual or Shared Sensor Networks.

In [21] the authors propose an optimization framework to maximize the Quality of Monitoring (QoM) in shared sensor networks. The proposed framework focuses on environmental monitoring applications whose reference "quality" can be modeled as dependent on the variance in the sensed data, and derives the application-to-sensors assignment which minimizes such variance. The same authors address in a later work the case where the application assignment problem is no longer centralized but rather distributed by resorting to game-theoretic tools [22]. Ajmal *et al.* leverage the concept of QoM and propose an admission control scheme to dynamically "admit" applications to be deployed on physical sensor networks. The authors of [23] focus on the problem of scheduling applications to shared sensor nodes with the ultimate goal of maximizing the sensor network lifetime. Along the same lines, Zeng *et al.* propose in [24] an optimization framework to prolong network lifetime by properly scheduling the tasks in a shared/virtual sensor network.

The problem of allocating physical resources to multiple applications is also often cast as an auction. In [25], the authors propose a reverse combinatorial auction, in which the sensor nodes act as bidders and bid cost values (according to their available resources) for accomplishing the subset of the applications' tasks. Optimal bidding strategies are then studied to make the auction effective and truthful.

To the authors' knowledge, this is the first approach to model and analyze the physical resource allocation problem (processing, storage, available bandwidth, limited communication range) for different applications in virtual sensor networks from an optimization point of view. This work extends and completes our previous work [26] with three main additional contributions: (i) we provide a complexity analysis of the proposed optimization problem, which is proven to be NP-complete; (ii) consequently, we introduce a heuristic iterative algorithm to obtain sub-optimal solutions of the resource allocation problem by reducing its complexity. We show in the performance evaluation section that, as the problem size grows, the heuristic algorithm provides results close to the optimum with a much lower computation time; (iii) a more comprehensive performance evaluation analysis of the proposed approach is carried out: in [26] we mainly focus on showing the benefits of virtualization. In this work we analyze thoroughly the impact of varying the main model parameters (number of scalar and multimedia nodes; number of sinks; lifetime; type of routing) in the system performance and we also evaluate the performance of the proposed heuristic algorithm.

3. System model and optimization framework

Table 1 summarizes the notation used through this section. Let $S = \{s_1, s_2, \ldots, s_l\}$ be a set of sensor nodes, $A = \{a_1, a_2, \ldots, a_m\}$ a set of applications which are to be deployed in the reference area, and $T = \{t_1, \ldots, t_n\}$ a set of test points in the reference network scenario. These test points are physical locations where the application's sensing parameter must be measured (e.g., a test point can be a physical location where a temperature monitoring application need to collect a temperature sample). To simplify notation, in the following we will use the subscript index i (or h) to refer to a sensor node s_i (or s_h), the subscript index j to refer to an application a_j and the subscript index k to refer to a test point t_k.

Each application j requires to sense a given set of test points $T_j \subseteq T$. Formally, the application j has to be deployed in a subset of sensor node set S such that all the test points in T_j are sensed. We consider that a test point is covered by a sensor node i if it is within its sensing range, R_i^s. Thus, given a test point, a set of sensor nodes can *cover* it (the test point can be in the sensing range of several nodes), but only one sensor node will *sense* it.

Therefore, it is convenient to introduce as well the set S_{jk} defined as the set of sensor nodes which cover the test point k, with $k \in T_j$. In other words, if the application j is deployed on any of the sensors in set S_{jk}, then the target test point k is sensed for

Please cite this article as: C. Delgado et al., On optimal resource allocation in virtual sensor networks, Ad Hoc Networks (2016), http://dx.doi.org/10.1016/j.adhoc.2016.04.004

Table 1
Set of parameters of the optimization framework.

Parameter	Definition
Propagation model	
P_{max}	Maximum transmission power
R^T_{max}	Maximum transmission range with P_{max}
R^I_{max}	Maximum interference range with P_{max}
g_{ih}	Channel gain from transmitter i to receiver h in the directional link (i, h)
d_{ih}	Distance from i to h
$R^T(p_i)$	Transmission range for node i with transmission power p_i
$R^I(p_i)$	Interference range for node i with transmission power p_i
Sets, vectors, revenues and costs	
$S = \{s_1, s_2, \ldots, s_l\}$	Set of sensor nodes; subscript index i (or h) refers to sensor node s_i (or s_h)
$SINK$	Set of sink nodes (a subset of S)
$A = \{a_1, a_2, \ldots, a_m\}$	Set of applications; subscript index j refers to application a_j
$T = \{t_1, \ldots, t_n\}$	Set of test points; subscript index k refers to test point t_k
T_j	Set of test points of application j
S_{jk}	Set of sensor nodes which cover test point k of application j
$r_j = \{c_j, m_j, l_j\}$	Requirement vector of application j (source rate, memory, processing load)
$o_i = \{C_i, M_i, L_i, E_i\}$	Resource vector of node i (bandwidth, storage, proccessing power, energy)
$Q = [q_1, q_2, \ldots, q_m]^T$	Preference vector across all the m applications
q_j	Revenue for having application j successfully deployed
δ_i	Cost incurred in activating sensor node i.
Coverage and application deployment	
z_j	Binary variable indicating if application j is successfully deployed
y_{ijk}	Binary variable indicating if test point k of application j is deployed at sensor node i
x_i	Binary variable indicating if sensor node i is active
h_{jk}	Binary variable indicating if test point k of set T_j is sensed by a sensor node
N_{ij}	Maximum number of test points of application j actually covered by sensor node i
Routing	
f_{ih}	Flow of data in bps transmitted from node i to node h
l_{ih}	Binary constant indicating if the distance between node i and node h is lower than R^T_{max}
b_{ih}	Binary variable indicating if data are transmitted from node i to node h
K	Constant higher than the maximum transmission rate of a node
Bandwidth	
C_{ih}	Capacity of the link (i, h)
IF_{ih}	Fraction of time that other links interfere the link (i, h)
Energy	
P^t_i	Power dissipation at the radio transmitter of node i
P^r_i	Power dissipation at the radio receiver of node i

this application. A necessary condition for an application j to be successfully deployed is that all the test points in its target set T_j must be sensed.

Each application j in A is further characterized by a requirement vector $r_j = \{c_j, m_j, l_j\}$ which specifies the generated source rate [bit/s], memory [bits] and processing load [MIPS] consumed by the application when it is deployed on a sensor node. The requirement vector can be interpreted as the amount of resources needed to accomplish the specific tasks required by the application (e.g., acquire, process and transmit 10 temperature samples, or acquire process and transmit one JPEG image, etc.). Additionally, each sensor node i in S is characterized by a given resource vector $o_i = \{C_i, M_i, L_i, E_i\}$, which specifies its available bandwidth, storage capabilities, processing power and energy store.

A protocol interference model with power control [27] is used to characterize the wireless communications among the sensor nodes. The maximum transmission power is $P_{\max}$. With this power, there are a maximum transmission range $R^T_{\max}$ and a maximum interference range $R^I_{\max}$. Given a directional link between a pair of nodes (i, h), the channel gain from transmitter i to receiver h is defined as $g_{ih} = g_0 \cdot d_{ih}^{-\gamma}$, being d_{ih} the distance from i to h, γ the path loss index and g_0 a constant dependent on antenna parameters. A transmission is successful if the received power exceeds a threshold α. Additionally, all the nodes under the interference range of a sensor node share the same transmission channel and therefore, the transmission time must be divided between them. If p_i is the transmission power assigned to node i, a transmission towards h is successful if $p_i \cdot g_{ih} > \alpha$. Thus, the transmission range for node i with transmission power p_i can be obtained as $R^T_i(p_i) = (p_i \cdot g_0/\alpha)^{1/\gamma}$. Similarly, the interference resulting from node i with power p_i is non-negligible only if it exceeds a certain threshold β. Then, the interference range is $R^I_i(p_i) = (p_i \cdot g_0/\beta)^{1/\gamma}$.

Qualitatively, the application assignment problem for virtual sensor networks can be defined as follows: to maximize the weighted number of deployed applications subject to coverage constraints (the set of test points of each application must be sensed) and application requirements (each application should be assigned enough bandwidth, and processing and storage resources to operate successfully). In addition, due to the multihop nature of WSNs, routing and link capacity constraints must be considered when the data generated by the application has to be delivered remotely.

Further, let us assume that a preference vector across all the m applications is defined, $Q = [q_1, q_2, \ldots, q_m]^T$, where q_j represents the revenue for the network provider for having application j successfully deployed in the network. Let z_j be a binary variable indicating if application j is successfully deployed in the network. Let y_{ijk} be a binary variable indicating if application j is deployed at sensor node i covering test point k. Let x_i be a binary variable indicating if sensor node i is active in the network. Let h_{jk} be a binary variable which indicates if test point k belonging to set T_j is sensed by a sensor node which runs application j.

Please cite this article as: C. Delgado et al., On optimal resource allocation in virtual sensor networks, Ad Hoc Networks (2016), http://dx.doi.org/10.1016/j.adhoc.2016.04.004

The objective function aims at maximizing the overall revenue out of the application deployment process while minimizing the cost related to activating sensor nodes:

$$\max\left(\sum_{j\in A} q_j z_j - \sum_{i\in S}\delta_i x_i\right) \tag{1}$$

where δ_i is the cost incurred in activating sensor node i.

3.1. Constraints on coverage and on resources of the sensors

Constraints (2)–(5) require that all the applications which are actually deployed do fulfill the coverage constraints, that is, they sense all the required test points. Specifically, Eq. (2) indicates that a test point k of an application j is sensed if the application j is deployed at a sensor node i belonging to S_{jk}. If so, it ensures that it is only sensed by sensor node i. Eq. (3) ensures that if a sensor i does not cover a test point k of an application j, then it can not sense that test point. Depending on the application, it can be possible that the same sensor node can sense several of its test points (e.g., visual applications). If we define N_{ij} as the maximum number of test points of the same application j that a sensor i is able to sense, Eq. (4) guarantees that this threshold is not exceeded. Eq. (5) indicates that an application j is successfully deployed, i.e., $z_j = 1$, only if all its test points are sensed ($h_{ij} = 1, \forall k \in T_j$). On the other hand, if the application is not successfully deployed, i.e., if $z_j = 0$, the constraint forces that none of its test points are sensed by any node ($h_{ij} = 0, \forall k \in T_j$ and consequently according to Eq. (2), $y_{ijk} = 0, \forall k \in T_j, \forall i \in S_{jk}$). This guarantees that if the application cannot be deployed, resources are not wasted.

$$\sum_{i\in S_{jk}} y_{ijk} = h_{jk} \qquad \forall j \in A, \forall k \in T_j \tag{2}$$

$$y_{ijk} = 0 \qquad \forall i \notin S_{jk}, \forall j \in A, \forall k \in T_j \tag{3}$$

$$\sum_{k\in T_j} y_{ijk} \le N_{ij} \qquad \forall i \in S, \forall j \in A \tag{4}$$

$$z_j = \frac{\sum_{k\in T_j} h_{jk}}{|T_j|} \qquad \forall j \in A \tag{5}$$

Constraints (6) and (7) are budget-type constraints for the available storage and processing load of the sensor nodes.

$$\sum_{j\in A}\sum_{k\in T_j} m_j y_{ijk} \le M_i \qquad \forall i \in S \tag{6}$$

$$\sum_{j\in A}\sum_{k\in T_j} l_j y_{ijk} \le L_i \qquad \forall i \in S \tag{7}$$

3.2. Routing constraints

The deployed applications will require most likely that the information generated locally is delivered remotely to collection points (sink nodes) through multihop paths. Note that these sensor nodes may run deployed applications or not. By resorting to a fluid model, it should be ensured that all the data produced by the sensors running applications are received by the sink nodes. This fact can be conveniently expressed using the following constraints:

$$\sum_{\substack{h\in S\\ i\neq h}} f_{hi} - \sum_{\substack{h\in S\\ h\neq i}} f_{ih} + \sum_{j\in A}\sum_{k\in T_j} c_j y_{ijk} = 0 \quad \forall i \in S \setminus SINK \tag{8}$$

$$\sum_{j\in A} |T_j| c_j z_j = \sum_{h\in SINK}\left(\sum_{\substack{i\in S\\ i\neq h}} f_{ih} + \sum_{j\in A}\sum_{k\in T_j} c_j y_{hjk}\right) \tag{9}$$

where $SINK$ is the set of sink nodes (a subset of S) and f_{ih} is a variable representing the flow of data in bps transmitted from node i to node h. Constraints (8) enforce flow conservation at sensor nodes: the incoming flow rate to a node i plus the data generated by itself must be equal to the outcoming flow rate. The variable y_{ijk} is 1 if sensor node i senses test point k of application j, i.e., if node i is running application j and therefore generating c_j bps because it is close enough to test point k to monitor the sensing parameter of application j. Constraint (9) imposes that the amount of data generated in the network is equal to the amount of data collected by the set of sinks. The left term represents the total data rate generated in the network: for each active application j ($z_j = 1$) there are $|T_j|$ sensor nodes sensing the corresponding $|T_j|$ test points and therefore each one generating c_j bps. The right term represents the total data rate received by the set of sink nodes plus the rate generated by themselves in case they were also running applications. This equality, together with the flow conservation in constraints (8), imposes that all the data rate generated in the network is finally collected by the set of sinks (delivered by other nodes or generated by the sinks themselves).

The following constraint set enforces that if a sensor node is either running an application or receiving data, then it must be active in the network:

$$\sum_{\substack{h\in S\\ h\neq i}} f_{hi} + \sum_{j\in A}\sum_{k\in T_j} c_j y_{ijk} \le K x_i \qquad \forall i \in S \tag{10}$$

where K is a constant high enough (higher than the maximum transmission rate of a node). Finally, constraints

$$f_{ih} \le K l_{ih} \qquad \forall i, h \in S \tag{11}$$

where l_{ih} is a constant that indicates if there is a viable link between i and h, i.e., if the distance between both nodes is less than the maximum transmission range R^T_{max}, then $l_{ih} = 1$ and $l_{ih} = 0$ otherwise. Therefore, these constraints ensure that data must be transmitted exclusively along neighboring nodes.

These expressions allow flow splitting and multipath routing. In the sequel, we will denote this kind of routing as *multipath routing*.

However, in WSNs routes from each sensor node to a sink node follow typically a single path, such as the Destination Oriented Directed Acyclic Graph (DODAG) of RPL [28]. Therefore, we introduce the following restrictions to ensure that all the traffic flowing out of a sensor has only one possible route to a sink:

$$b_{ih} \le l_{ih} \qquad \forall i, h \in S \tag{12}$$

$$\sum_{h\in S} b_{ih} \le 1 \qquad \forall i \in S \tag{13}$$

$$f_{ih} \le K b_{ih} \qquad \forall i, h \in S \tag{14}$$

where b_{ih} is a binary variable which indicates if data are transmitted from node i to node h. Constraints (13) and (14) impose that only one link from sensor node i to any of its neighbors transports all the data that i must forward. In the sequel, we will denote this kind of routing as *singlepath routing*.

Including the route creation in the optimization framework may not be always feasible. In addition, since all the traffic in WSN is forwarded to a single or a limited number of sinks, the main bottleneck will be mainly the last hop to these sinks. For these reasons, we also consider the possibility of excluding the routing from the optimization process and assuming a predefined set of routes from each node to a sink. To that purpose, we build DODAGs using the number of hops as a metric (i.e., when there are several sinks, each node belongs to the DODAG that reaches a sink with the minimum number of hops). This implies that Eqs. (12)–(14) must be excluded from the model and that for each node i, the constant l_{ih} is 1 just for a single h (the father node in the routing tree towards

Please cite this article as: C. Delgado et al., On optimal resource allocation in virtual sensor networks, Ad Hoc Networks (2016), http://dx.doi.org/10.1016/j.adhoc.2016.04.004

the sink) and therefore $f_{ih'} = 0$ for all $h' \neq h$. In the sequel, we will denote this kind of routing as *static routing*.

3.3. Bandwidth constraints

The available bandwidth in the network is limited and must be shared among sensor nodes. We assume that a fair medium access control scheme orchestrates the access to the shared medium. Given a directional link between a pair of nodes (i, h), let the capacity of the link be defined as $C_{ih} = \min(C_i, C_h)$. This aims to model that the transmission rate is limited by the most restrictive node in the link. Transmissions of other links where i or h are either transmitter or receiver cannot be simultaneously active with (i, h) (note that some combinations are not possible in this particular case due to routing constraints, i.e., another link with i as a transmitter).

According to the considered protocol interference model, the interfering links for link (i, h) are those whose receiver is within the interference range of node i or the links where h is within the interference range of its transmitter. Although none of these links can be simultaneously active with (i, h), some of them (depending on their relative positions) could be simultaneously active with each other. Therefore, if we define IF_{ih} as the fraction of time that other links interfere the link (i, h), we have that:

$$IF_{ih} = \sum_{\substack{g \in S \\ g \neq h}} \frac{f_{ig}}{C_{ig}} + \sum_{g \in S} \frac{f_{gi}}{C_{gi}} + \sum_{\substack{g \in S \\ g \neq i}} \frac{f_{hg}}{C_{hg}} + \sum_{\substack{g \in S \\ g \neq i}} \frac{f_{gh}}{C_{gh}} + \sum_{\substack{g,t \in S \\ d_{it} < R_i^I(p_i)}} \frac{f_{gt}}{C_{gt}} + \sum_{\substack{g,t \in S \\ d_{gh} < R_g^I(p_g)}} \frac{f_{gt}}{C_{gt}} \quad (15)$$

Then, for each link (i, h) in the network it must be ensured that the fraction of time used by the link plus all its interferences is less or equal to 1:

$$\frac{f_{ih}}{C_{ih}} + IF_{ih} \leq 1 \qquad \forall i, h \in S \quad (16)$$

Constraints (16) are the equivalent budget-type constraints for the available wireless capacity to the storage and processing load constraints given in (6) and (7).

3.4. Energy constraints

Finally, energy constraints are included to ensure that the application deployment pattern guarantees a minimum lifetime L for the virtual sensor network. Typically, energy consumption due to wireless communication (i.e., transmitting and receiving) has been considered the dominant factor in power consumption for WSNs [29]. While this is the case for traditional scalar applications, where processing is limited to simple operations, in multimedia applications the energy required to process data can not be neglected [30].

Regarding wireless transceiver, the power dissipation at the radio transmitter P_i^t or at the radio receiver P_i^r of each node i can be modeled as [31]:

$$P_i^t = \sum_{h \in S, h \neq i} \left(\beta_1 + \beta_2 d_{ih}^{\gamma}\right) f_{ih} \qquad \forall i \in S \quad (17)$$

$$P_i^r = \rho \sum_{h \in S, h \neq i} f_{hi} \qquad \forall i \in S \quad (18)$$

Typical values for β_1, β_2 and ρ are $\beta_1 = \rho = 50$ nJ/bit and $\beta_2 = 0.0013$pJ/bit/m^4, with $\gamma = 4$ the path loss index.

The estimation of the power dissipation due to the processing load is not so straightforward, since it depends on several factors such as the hardware architecture of the nodes or the specific implementation of the algorithm for each application. In the lifetime constraints set in (19), this power dissipation is left as a function f of the processing loads l_j of the applications. In Section 5, further details about the specific evaluated multimedia applications are given.

$$P_i^t + P_i^r + f\left(\sum_{j \in A} \sum_{k \in T_j} y_{ijk} l_j\right) \leq \frac{E_i}{L} \qquad \forall i \in S \quad (19)$$

3.5. Complexity analysis

Theorem 1. *The application deployment problem is NP-complete.*

Proof. The NP-completeness can be proved by restriction, that is by showing that our general application deployment problem contains a known NP-complete problem as a special case [32]. The reference problem we use in the proof is the multiple knapsack problem which is known to be NP-complete. The proof is based on specifying the additional restrictions to be added to the application deployment problem so that the resulting restricted problem will be identical to the multiple knapsack problem, which is defined and explained next:

The multiple knapsack problem is a well-known problem in combinatorial optimization. There is a set N of items, with each item $j \in N$ having an associated profit p_j and weight w_j. In addition, there is a set M of knapsacks, each one $i \in M$ having capacities W_i. A binary decision variable x_{ij} is used to select the item j to knapsack i. The objective is to pick some of the items, with maximal total profit, while the maximum total weight of the chosen items at each knapsack does not exceed its capacity W_i, i.e.:

$$\max \sum_{j \in N} \sum_{i \in M} x_{ij} p_j$$

s.t.

$$\sum_{i \in M} x_{ij} \leq 1 \qquad \forall j \in N$$

$$\sum_{j \in N} w_j x_{ij} \leq W_i \qquad \forall i \in M$$

In order to show that this multiple knapsack problem reduces to our application deployment problem, let's consider the particular instance of the application deployment problem characterized by the following setting: $\delta_i = 0$, $\forall i \in S$ (the cost incurred in activating sensor node i, as defined in the objective function, is negligible), $S_{jk} = S$, $\forall j \in A$ (all the test points are reachable from all the sensor nodes), and $T_j = \{1\}$, $\forall j \in A$ (all the applications need to sense one single test point). This last assumption also implies that $N_{ij} = 1$ (the maximum number of test points of the same application j that a sensor i is able to sense is also 1). Let's further assume that routing is not needed, that is, formally $SINK = S$, and that sensor nodes do not have processing capability limitation, $L_i = \infty$, $\forall i \in S$. In such setting, since all the applications need to sense one single test point which is "reachable" from all the sensor nodes (from $T_j = \{1\}$ and $S_{jk} = S$), the index k can be safely dropped from variables y_{ijk} and h_{jk}. The application deployment problem can be re-written as follows:

$$\max \sum_{j \in A} q_j z_j$$

s.t.

$$\sum_{i \in S} y_{ij} = h_j \qquad \forall j \in A$$

$$y_{ij} \leq 1 \qquad \forall i \in S, \forall j \in A$$

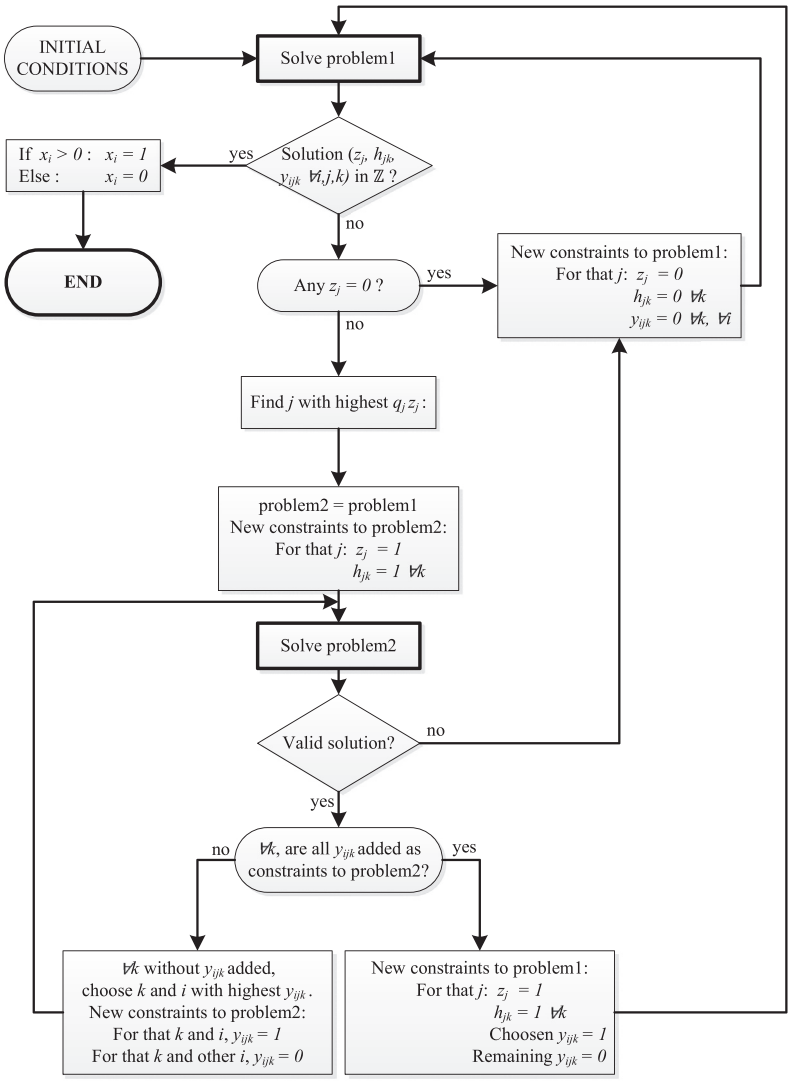


Fig. 1. Diagram of the heuristic algorithm.

$$z_j = h_j \qquad \forall j \in A$$

$$\sum_{j \in A} m_j y_{ij} \leq M_i \qquad \forall i \in S$$

This formulation can be further simplified: from first and third constraints we have $\sum_{i \in S} y_{ij} = z_j$ and h_j can be removed from the problem. Since z_j is a binary variable, we can re-write again this expression as $\Sigma_{i \in S} y_{ij} \leq 1$, substitute z_j by $\Sigma_{i \in S} y_{ij}$ in the objective function and remove z_j from the problem. Finally, the second constraint $y_{ij} \leq 1$ is already contained in $\Sigma_{i \in S} y_{ij} \leq 1$ and can be also safely removed, which leads to:

$$\max \sum_{j \in A} \sum_{i \in S} y_{ij} q_j$$

s.t.

$$\sum_{i \in S} y_{ij} \leq 1 \qquad \forall j \in A$$

$$\sum_{j \in A} m_j y_{ij} \leq M_i \qquad \forall i \in S$$

This last formulation matches the multiple knapsack problem previously defined, which is known to be NP-complete. By restriction, also the application deployment problem must be NP-complete [32]. □

4. Heuristic algorithm

The resource allocation problem presented in Sections 3.1–3.4 is a Mixed Integer Linear Programming (MILP) problem which has been shown to be NP-complete in Section 3.5. As a consequence, the time required to solve the problem increases very quickly as the size of the problem grows and therefore it may be not scalable, as it will be later shown in Section 5.5. Consequently, in this section we propose a heuristic iterative algorithm to obtain suboptimal solutions to the problem in reduced computation time. The algorithm follows a classical approach based on the Linear Programming (LP) relaxation of the original MILP problem [33,34].

In short, the algorithm is based on the iterative resolution of simplified relaxed LP problems. The formulation of these problems focuses on the *static routing* strategy defined in Section 3.2 and drops the integrality constraints on variables z_j, x_i, y_{ijk} and h_{jk}. In each iteration step, the validity of the obtained solution to be feasible in the original MILP problem is checked to stop the algorithm. Fig. 1 shows the diagram block of the algorithm, whose steps are explained next.

The algorithm iteratively solves a relaxed LP problem, named *problem1*, modified with new constraints in each step until a valid solution is found. In addition, in some steps it is necessary to sub-

Please cite this article as: C. Delgado et al., On optimal resource allocation in virtual sensor networks, Ad Hoc Networks (2016), http://dx.doi.org/10.1016/j.adhoc.2016.04.004

iteratively solve a new temporary problem, named *problem2*, to find the proper constraints to be added in *problem1*.

Problem1 initially corresponds to the relaxed original problem, i.e. the resource allocation problem with variables z_j, x_i, y_{ijk} and h_{jk} taking any real value between 0 and 1. As the algorithm progresses, in each iteration step new constraints are added to force an application to be added or excluded until a decision is made for the whole set of applications.

In any step, if the solution of *problem1* corresponds to integer values for all the relaxed variables but x_i, then a valid solution is found and the algorithm is stopped. Then, the final solution for the original MILP problem only requires to set to 1 all the x_i variables that are positive (i.e. we set any $x_i > 0$ to $x_i = 1$). This can be done since the only constraint where variables x_i are involved is constraint (10), which will be always fulfilled when rounding $x_i > 0$ to $x_i = 1$. On the other hand, new constraints are added when *problem1* does not have an integer solution. First, we see if there are applications that are not active at all in the solution ($z_j = 0$) and choose one of them at random. Then, we force this application to be inactive by making $z_j = 0$, $y_{ijk} = 0\ \forall k$, $\forall i$ and $h_{jk} = 0\ \forall k$. We follow this procedure until $z_j > 0$ for all the remaining applications.

Then, we try to find the set of constraints to force an application to be activated. For that, we choose the application j with maximum value of $q_j z_j$. We then define the temporary problem *problem2* equal to *problem1* in the current iteration step and add the constraints to force the application to be active ($z_j = 1$ and $h_{jk} = 1$ for all the test points of this application). If *problem2* does not have a feasible solution, we dismiss *problem2*, add to *problem1* the constraints to make this application j inactive and jump to the next step. On the other hand, if *problem2* does have a feasible solution, the application could be activated, but the integer values of y_{ijk} still have to be obtained: the temporary solution does not guarantee that each test point k is sensed by a single node i ($y_{ijk} = 1$) as the variables y_{ijk} are still relaxed. To set these integer values we proceed as follows: iteratively for each test point k, new constraints for the variables y_{ijk} are added to *problem2*, solving this problem at most $|T_j|$ times (the number of test points k of application j). Specifically, in each sub-step we look for the highest relaxed y_{ijk} (we will obtain which is the node i that better covers the test point k). For that test point k, we force the corresponding node i to sense it completely adding to *problem2* the constraint $y_{ijk} = 1$. For the rest of the nodes, we add the $y_{ijk} = 0$ constraints in order to avoid the waste of resources. Any time the *problem2* is solved, if there is not a feasible solution, this temporary sub-iteration is stopped: *problem2* is dismissed, the constraints to make the application j inactive are added to *problem1* and we jump to the next step. On the contrary, if *problem2* finds a valid solution when all the constraints for y_{ijk} have been added, all these temporary constraints (related to z_j, h_{jk} and y_{ijk}) already added to *problem2* to activate the application j are added to *problem1* and the next step follows.

This process is repeated until the solution of *problem1* fulfills the integrality of variables z_j, h_{jk} and y_{ijk}. At this moment, as stated before, variables x_i are still real, but the original MILP problem is guaranteed to have a solution, since constraint (10) will be always fulfilled when we set $x_i = 1$ if the solution of *problem1* is $x_i > 0$ and 0 otherwise.

5. Performance evaluation

In this section, we evaluate the benefit of optimally deploying applications in general purpose, shared sensor networks with respect to vertical application-optimized sensor network deployments. To this extent, we compare the cases where physical resources (sensor networks) are *a priori* allocated to specific applications with the case where the same physical infrastructure is optimally shared among multiple applications according to our proposed optimization framework.

Table 2
Reference sensor node platforms.

	Basic	High-Level
Reference Hardware	Telos-B	BeagleBoard
TX Rate (C_i)	250[kb/s]	250[kb/s]
Available RAM (M_i)	7[Kbyte]	256[Mbyte]
Processing Rate (P_i)	8[MIPS]	720[MIPS]
Energy Store (E_i)	32400[J]	32400[J]

Unless differently specified, the following results have been obtained by solving the optimization model of Section 3 using CPLEX software [35] in the reference test environment with two types of sensor node hardware, and four applications to be deployed. Results have been obtained averaging the outcome of the optimization over 100 realizations. Sensor nodes are deployed in a 200 × 200 m scenario. We consider a default sensing range of $R_i^s = 30$ m for all of the sensors [36]. A two-ray ground path loss model with $\gamma = 4$ and $g_0 = 8.1 \cdot 10^{-3}$ [37] is considered. P_{max} is set to 0 dBm and the receiver sensitivity α is −92 dBm [38], which implies a maximum transmission range R_{max}^T of 59 m. Similarly, the interference sensitivity is −104 dBm, which implies a maximum interference range R_{max}^I of 118 m.

In the following we first describe in details the reference playground in terms of sensor node hardware and reference type of applications, moving then to the assessment of the performance evaluation. It is worth pointing out here that the proposed optimization framework is general and can be applied to any network topology setting and application requirements.

5.1. Sensor nodes

To reflect the inherent heterogeneity of sensor nodes hardware platform in common Internet of Things applications, the network topologies used for performance evaluation include sensor node hardware with different characteristics in terms of cost and available resources; namely, we consider *basic* highly-constrained hardware and *high-level* sensor node hardware. The parameters of *basic* sensor nodes have been derived by taking as a reference commercial platforms like the TelosB sensor motes [39], whereas the *high-level* sensor nodes are well represented by BeagleBone platforms [40] or similar. Table 2 summarizes the main characteristics of the reference sensor node platforms used in the performance evaluation.

Each *basic* sensor node is assumed to mount a temperature and a light sensor, an IEEE 802.15.4 radio with integrated antenna and a 8[MHz] TI MSP430 microcontroller which can operate at 8[MIPS] and with 10[KB] RAM, although only 7[KB] are available for applications [17]. Therefore, the corresponding resource vector for basic nodes is $o_i = \{C_i, M_i, L_i, E_i\} = \{250$ kb/s, 7 KB, 8 MIPS, 32400 J$\}$.

The *high-level* sensor nodes are assumed to have a 720[MHz] super-scalar ARM Cortex-A8 processor (up to 720[MIPS]) and 256[MB] of RAM. High-level nodes also have a IEEE 802.15.4-compliant transceiver, a low-power USB camera for multimedia applications and also scalar sensors. The reference resource vector for *high-level* nodes is $o_i = \{250$ kb/s, 256 MB, 720 MIPS, 32400 J$\}$.

The energy budget for both nodes is 32400[J] assuming that a node runs at 3[V] with 3[Ah] of battery supply (2 AA batteries).

5.2. Reference applications mix

In the performance evaluation we focus on two classes of applications which are to be deployed: *scalar* applications, which re-

Table 3
Reference applications.

	Scalar		Visual	
	Temperature Mon.	Light Mon.	CTA	ATC
Generated Data (c_j)	0.5[kb/s]	1[kb/s]	20[kb/s]	12[kb/s]
Bytecode footprint (m_j)	4462[byte]	1006[byte]	10[kbyte]-256[Mbyte]	10[kbyte]-256[Mbyte]
Processing load (l_j)	negligible	negligible	17.64[MIPS]	69.23[MIPS]

quire the sensing and delivery of scalar information, and *multimedia* applications, which require the sensing, processing and delivery of multimedia content (images and video). Table 3 summarizes the characteristics of the reference applications used in the performance evaluation.

As for *scalar* applications, we consider temperature and light monitoring applications characterized by the following parameters: the bytecode of temperature and light monitoring applications is 4462 bytes and 1006 bytes, respectively as specified in [17]; the reference sampling rate is 0.5[Hz] for temperature and 1[Hz] for light monitoring, which are consistent to typical sampling rate for scalar applications [41]. Considering a packet size of 127 bytes per sample, the temperature application has a source rate of 0.5[kb/s], whereas for the light application is 1[kb/s]. The processing load l_j required by these type of applications is considered negligible [1].

Thus, the requirement vector for temperature monitoring is $r_j = \{c_j, m_j, l_j\} = \{$ 0.5 kb/s, 4462 B, -$\}$ whereas for light monitoring is $r_j = \{$1 kb/s, 1006 B, -$\}$

For *multimedia* applications we focus on visual sensor networks, i.e. WSNs designed to perform visual analysis (e.g. object recognition) [30]. We consider two paradigms to perform visual tasks: the classic Compress-Then-Analyze (CTA) and the Analyze-Then-Compress (ATC) [30,42]. In CTA-based multimedia applications, images are acquired from camera nodes, which compress them locally (as an example using JPEG compression) and send the compressed information to a central controller, which implements the visual task at hand.

On the other hand, in ATC-based multimedia applications, the camera nodes pre-process the acquired image locally by extracting distinctive local or global visual features capturing the "most" important visual content in the original image. Visual features, instead of the pixel-level representation of the image, are then sent remotely to the central controller for further analysis.

In [42] a detailed characterization of transmission rates and energy consumption for both approaches is provided. Based on this analysis, the following requirements vectors are used to represent visual applications based on CTA and ATC paradigms, respectively: $r_j = \{$20 kb/s, 10 KB $< m_j <<$ 256 MB, 17.64 MIPS$\}$, $r_j = \{$12 kb/s, 10KB $< m_j <<$ 256 MB, 69.23 MIPS$\}$. Details on how these numbers have been derived are reported in Appendix A.

5.3. Assessing the gain of virtualization

We start off by evaluating the benefit of having a shared infrastructure optimally orchestrated rather than multiple separated physical networks tailored to specific applications. To this extent, we consider two cases: (i) two separated wireless sensor networks coexist in the same geographic area to support scalar and multimedia application, respectively; (ii) a general purpose wireless sensor network is shared among scalar and multimedia applications. Case (i) consists of a sensor network formed by 36 TelosB-like to support scalar applications (temperature and light monitoring), and a sensor network formed by 36 BeagleBone nodes to support visual applications (CTA-based and ATC-based); in case (ii), the same 36 TelosB-like and 36 BeagleBone-like nodes are used as a single physical infrastructure, which can be optimally shared among scalar and multimedia applications according to the virtualization framework proposed in Section 3.

The following results have been obtained by setting the number of test points to be sensed to 5 for the scalar applications and 3 for the visual ones. The positions of all the test points are randomly generated in the considered scenario. We further assume that each sensor is able to sense $N_{i,j} = 1$ test points of the same application and that each network has a sink node (one of the 36 nodes). The minimum lifetime for the virtual sensor network is $L = 1$ day. Similar results have been obtained under different parameter setting.

Figs. 2 and 3 show the performance of both networks in terms of the number of active applications and the number of active nodes when the WSNs work isolated and also when the 72 nodes cooperate as a single network that gives support to all the applications. The curves labeled as "Scalar" and "Visual" have been obtained by applying the optimization problem separately to the two separated wireless sensor networks of the aforementioned case (i); conversely, the curves labeled as "Joint" have been obtained by applying the optimization problem to the network of case (ii); the "Joint"-labeled curves are further distinguished with respect to the preference vectors, Q, assigned to the application mix: P1 refers to the case where all the applications have the same weight, P2 refers to the case where the different applications have different weights, namely scalar applications "weigh" 1, multimedia ATC-based applications weigh 8, and multimedia CTA-based applications weigh 12. Q represents the revenue for the network provider for having applications successfully deployed in the network. We have made the assumption that the higher the resource demands of an application are, the higher the benefit the provider obtains when deploying it should be. Therefore, since the main limiting factor for visual applications with regard to scalar ones is the bandwidth, preference values are approximately adjusted according to the demanded bandwidth: preference for scalar applications is 1 (they need 2.5 or 5 kb/s $= 0.5$ or 1 kb/s per test point $\times$ 5 test points); preference for ATC applications is 8 (they require 36 kb/s $= 12$ kb/s $\times$ 3 test points); and preference for CTA applications is 12 (they need 60 kb/s $= 20$ kb/s $\times$ 3 test points).

For each point in the curves, the same number of applications of each type (temperature, light, CTA and ATC) is generated, which is the value shown in the x-axis. For example, a 2 value in the x-axis represents a scenario where 2 temperature, 2 light, 2 CTA and 2 ATC applications try to be deployed.

The main message coming from Figs. 2(a) and 3(a) is that there is a clear gain both in terms of total deployed applications (higher) and in terms of total activated sensor nodes (lower) when the physical network infrastructure is optimally shared (case(ii)) with respect to the case where "vertical" sensor networks are independently optimized (case (i)).

[1] i.e. the energy required for acquiring and processing one sample is assumed to be much lower than the energy for transmitting it remotely

Please cite this article as: C. Delgado et al., On optimal resource allocation in virtual sensor networks, Ad Hoc Networks (2016), http://dx.doi.org/10.1016/j.adhoc.2016.04.004

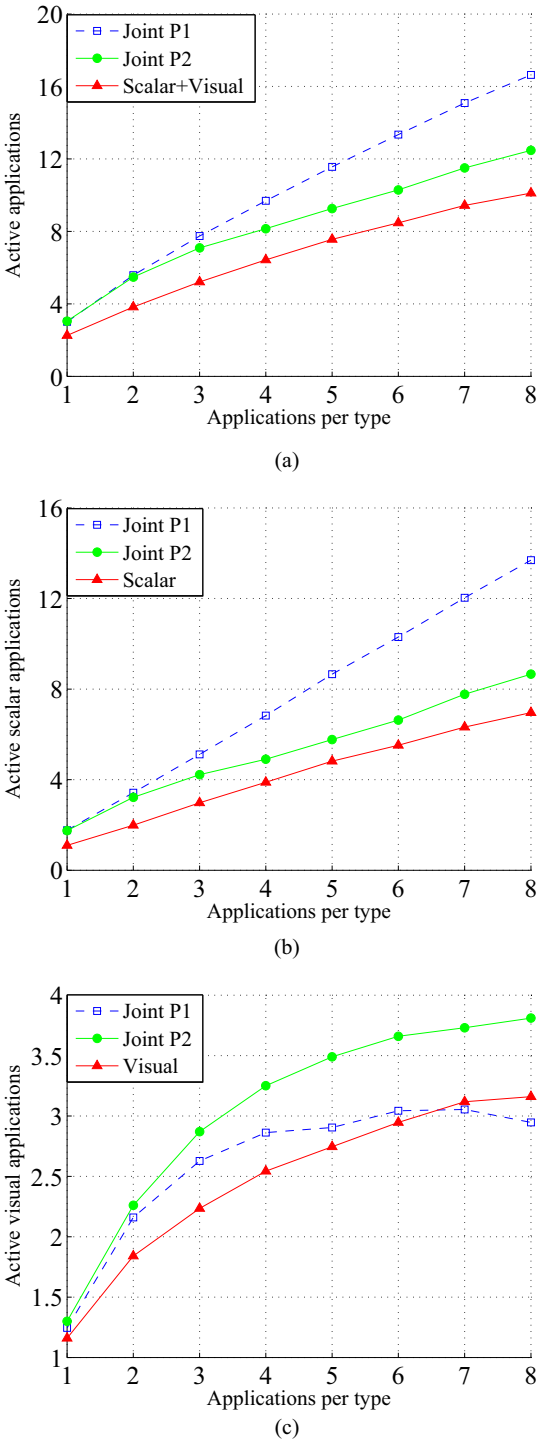


Fig. 2. Number of active applications vs offered applications per type. (a) Total (b) Scalar (c) Visual.

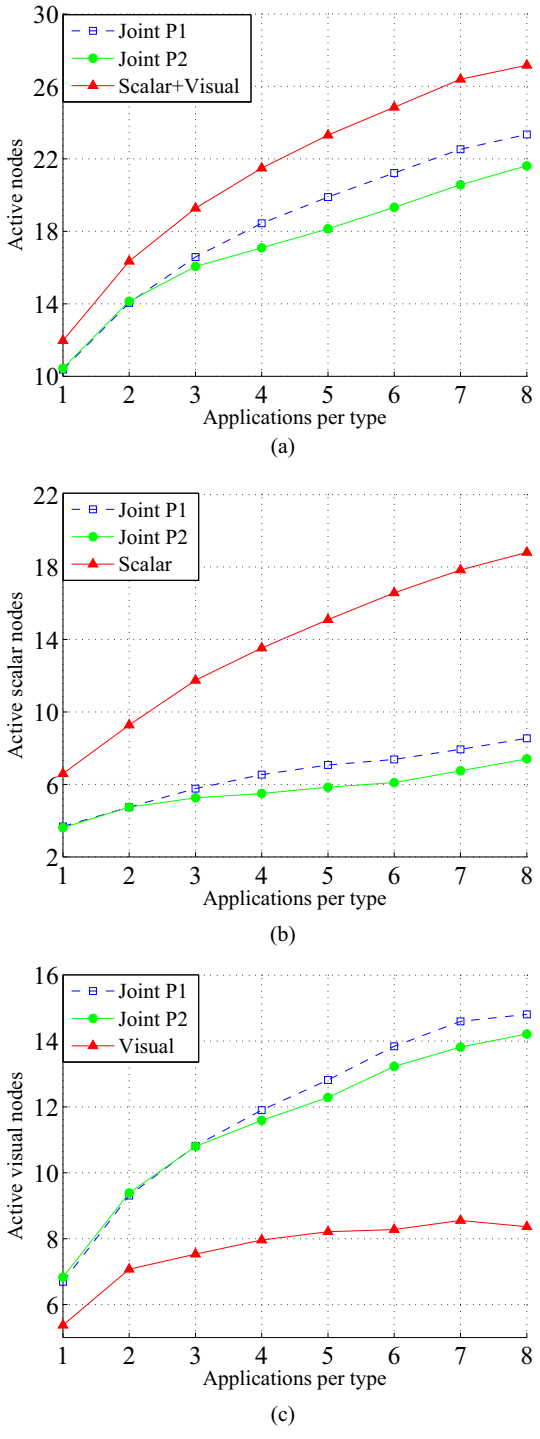


Fig. 3. Number of active nodes vs offered applications per type. (a) Total (b) Scalar (c) Visual.

Figs. 2 (b) and 2(c) report the breakdown of the deployed applications distinguished between scalar and visual ones. Expectedly, when all the applications have the same weights (homogeneous preference vector, Q), the optimal solution in the shared case tends to favor the deployment of higher numbers of more lightweight scalar applications, rather than lower numbers of "more heavy" visual applications.

Regarding the type of active nodes, Fig. 3(c) shows that the amount of active multimedia nodes increases for the joint scenario. The reason being that is multimedia nodes can also be used to host scalar applications.

5.4. Sensitivity of the virtualization gain

In this section, we evaluate how the gain of virtualization varies under different parameter setting including the network scale (Section 5.4.1), the number of sinks (Section 5.4.2), the target network lifetime (Section 5.4.3) and type of routing paradigm (Section 5.4.4). Unless differently specified, the reference evalua-

Please cite this article as: C. Delgado et al., On optimal resource allocation in virtual sensor networks, Ad Hoc Networks (2016), http://dx.doi.org/10.1016/j.adhoc.2016.04.004

Table 4
Total number of deployed applications and value of the optimization objective function (1) when varying the mix of sensor nodes available in the network. Column (a) reports the reference mix of sensor nodes in the format number of *Scalar Nodes (SN)-number of Visual Nodes (VN)*; column (b) reports the number of "offered" applications.

		SEPARATE		JOINT	
(a)	(b)	Total applications	Objective function	Total applications	Objective function
0SN-72VN	2	2.35	23.28	5.83	32.81
	4	2.97	29.21	8.20	43.12
	6	3.33	32.67	11.23	50.32
	8	3.52	34.58	12.73	53.91
24SN-48VN	2	2.93	21.61	5.75	29.36
	4	4.29	28.58	8.92	42.21
	6	5.17	32.48	10.75	47.86
	8	5.96	35.23	12.70	51.40
36SN-36VN	2	3.83	19.83	5.43	25.44
	4	6.43	27.84	8.15	36.88
	6	8.47	33.06	10.27	42.18
	8	10.12	36.39	12.47	45.48
48SN-24VN	2	3.80	13.86	4.85	19.76
	4	6.85	22.37	7.93	30.70
	6	9.63	29.09	11.00	36.28
	8	12.00	34.69	13.55	39.54
72SN-0VN	2	3.33	3.20	3.33	3.20
	4	6.75	6.56	6.75	6.56
	6	10.20	9.94	10.20	9.94
	8	13.38	13.04	13.38	13.04

tion environment includes 72 sensor nodes, 2 collection sinks and preference vector P2.

5.4.1. Impact of the number of nodes

First, we vary the composition of the types of nodes in the physical network, while maintaining the total number of nodes to 72. Table 4 reports the total number of deployed applications and the corresponding value of the optimization objective function (1) under different mixes of physical nodes for the two approaches (separate wireless networks vs joint wireless network). As clear form the table, the proposed optimal virtualization framework always leads to a considerable gain both in terms of total number of deployed applications and in terms of the objective function value.

Figs. 4 and 5 provide more insight on the analysis by reporting the breakdown of the deployed applications and the number of activated sensor nodes distinguished between scalar and visual ones. Expectedly, analyzing Fig. 4(b) for a fixed number of offered applications, the total number of deployed visual applications increases with the number of visual nodes in the reference topology, growing from 0, until its maximum value (when the 72 motes are multimedia); this is mainly due to the fact that the higher the number of visual nodes in the reference topology the higher is the probability that the offered visual application can be actually covered.

Similarly, going along the *x*-axis of Fig. 4(b) it can be noted that the number of deployed visual applications grows with the number of offered applications reaching a saturation value which is due to the bandwidth constraints in the optimization framework. On the other hand, the number of deployed scalar applications (Fig. 4(a)) does not seem to be much impacted by the number of available scalar nodes in the reference topology, since scalar application can be deployed also on visual nodes.

5.4.2. Number of sinks

Since the number of deployed applications depends on the bottleneck access to the sink, it is worth assessing the impact of the number of available sinks in the reference topology.

To this extent, Table 5 reports the total number of deployed applications and the reference value of the optimization objective function (1) for different values of the number of available sinks in

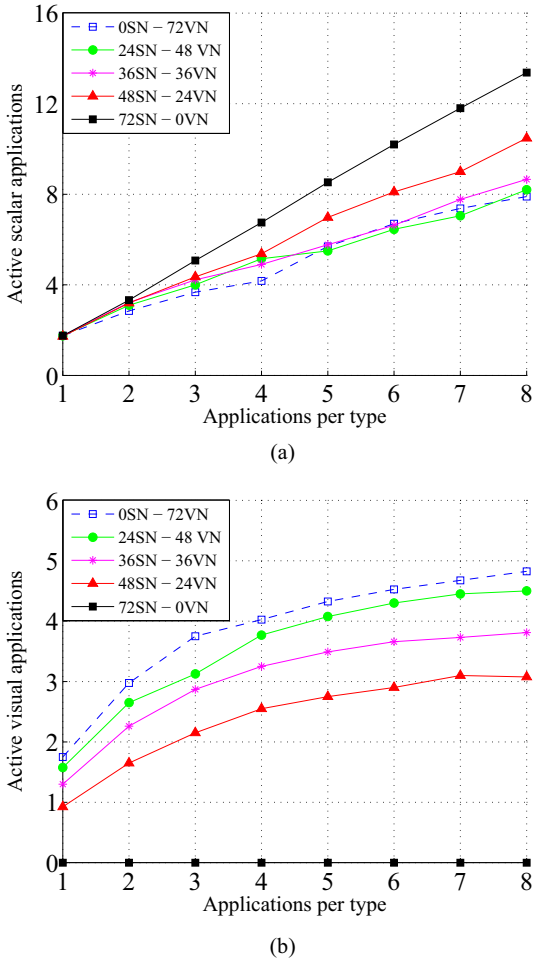
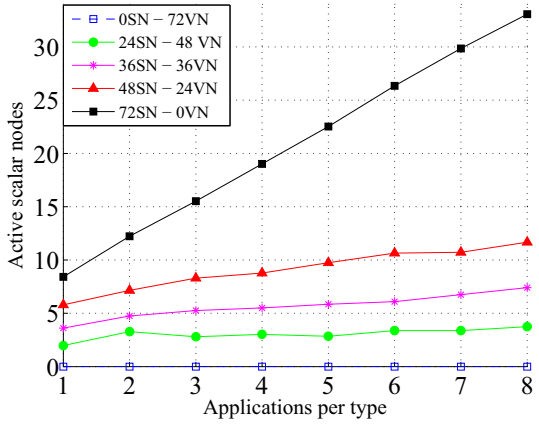


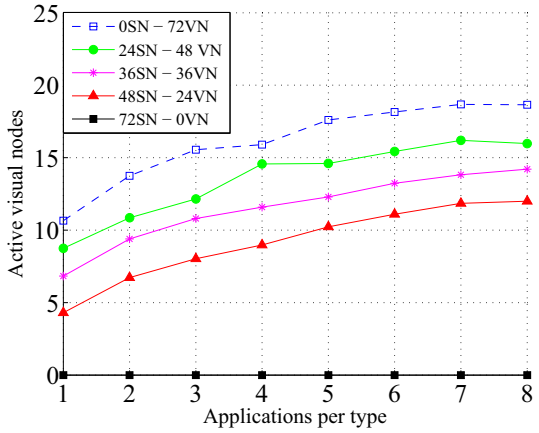
Fig. 4. Number of active applications vs offered applications per type varying the type of nodes. (a) Scalar (b) Visual.

Table 5
Total number of deployed applications and value of the optimization objective function (1) when varying the number of available information sinks. Column (a) reports the total number of available sinks; in the "separate" case half of the sinks specified here is dedicated to service each application type (e.g., if the number of sinks is equal to 4, it means that two sinks are dedicated for scalar applications and two sinks are dedicated to visual applications; column (b) reports the number of "offered" applications.

		SEPARATE		JOINT	
(a)	(b)	Total applications	Objective function	Total applications	Objective function
2	2	3.83	19.83	5.35	25.66
	4	6.43	27.84	8.18	36.83
	6	8.47	33.06	10.48	41.56
	8	10.12	36.39	12.73	44.79
4	2	4.02	22.51	5.63	28.19
	4	6.78	32.94	9.30	45.82
	6	8.85	38.90	12.23	53.19
	8	10.57	43.32	15.33	58.27
8	2	4.33	25.42	5.90	28.99
	4	7.44	39.81	11.15	53.13
	6	9.80	48.68	14.53	66.85
	8	11.72	54.98	17.48	75.46



(a)



(b)

Fig. 5. Number of active nodes vs offered applications per type varying the type of nodes. (a) Scalar (b) Visual.

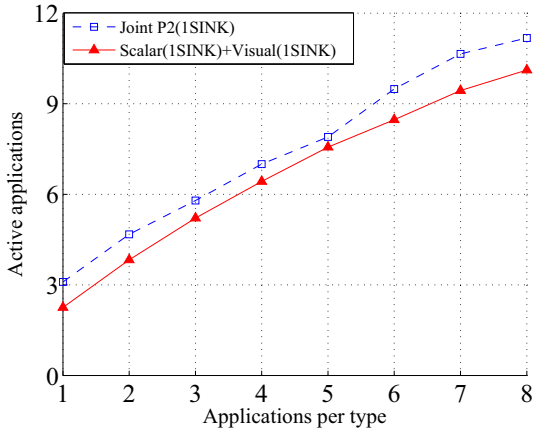


Fig. 6. Number of applications in case only 1 sink is available for the joint scenario.

the reference topology and for the two approaches (separate wireless networks vs joint wireless network). These numbers have been obtained by randomly drawing the corresponding number of sinks in the reference topology and averaging the outcome of the optimization over 100 realizations. As can be observed, there is a general improvement in the number of active applications as the number of sinks is increased. Still, the proposed optimal virtualization frameworks always lead to a considerable gain both in terms of total number of deployed applications and in terms of the objective function value.

Moreover, it is worth comparing the two cases where the wireless sensor networks are separate and each one of those has its own sink (two sinks in total) against the case where the two networks are completely shared with a single sink. Fig. 6 reports the total number of deployed applications in these two cases. Notably, even if the shared case has one single sink, and thus a lower bottleneck capacity, still the virtualization framework provides better results with respect to the case where two application-specific sinks can be used with double bottleneck capacity.

Figs. 7 and 8 provide application-specific results as the number of sinks is varied. As clear from Fig. 7, since visual applications are those with more demanding bandwidth requirements, they are the ones that benefit more of an increase in the number of sinks. On the other hand, scalar applications are less sensitive to the number of sinks: there is also an increase in the number of active scalar applications, but it is not so relevant due to their lower bandwidth requirements. Consequently, the number of active scalar nodes (Fig. 8(a)) does not depend on the number of sinks, whereas the number of active multimedia nodes (Fig. 8(b)) increases with the number of sinks since the number of active visual applications grows.

Please cite this article as: C. Delgado et al., On optimal resource allocation in virtual sensor networks, Ad Hoc Networks (2016), http://dx.doi.org/10.1016/j.adhoc.2016.04.004

Table 6
Total number of deployed applications and value of the optimization objective function (1) when varying the minimum lifetime constraint in the reference wireless sensor network. Column (a) reports the reference minimum lifetime; column (b) reports the number of "offered" applications.

		SEPARATE		JOINT	
(a)	(b)	Total applications	Objective function	Total applications	Objective function
1 hour	2	3.86	20.67	5.41	24.84
	4	6.45	29.07	8.45	38.20
	6	8.46	34.15	11.28	42.66
	8	10.12	38.05	13.73	46.41
1 day	2	3.83	19.83	5.43	25.44
	4	6.43	27.84	8.15	36.88
	6	8.47	33.06	10.27	42.18
	8	10.12	36.39	12.47	45.48
2 days	2	3.07	14.80	4.45	17.25
	4	5.29	21.05	8.03	27.66
	6	7.19	25.88	10.18	34.46
	8	8.81	29.79	12.85	39.04
8 days	2	1.99	1.90	3.33	3.21
	4	3.84	3.71	6.80	6.64
	6	5.47	5.31	10.30	10.11
	8	6.88	6.69	13.63	13.41

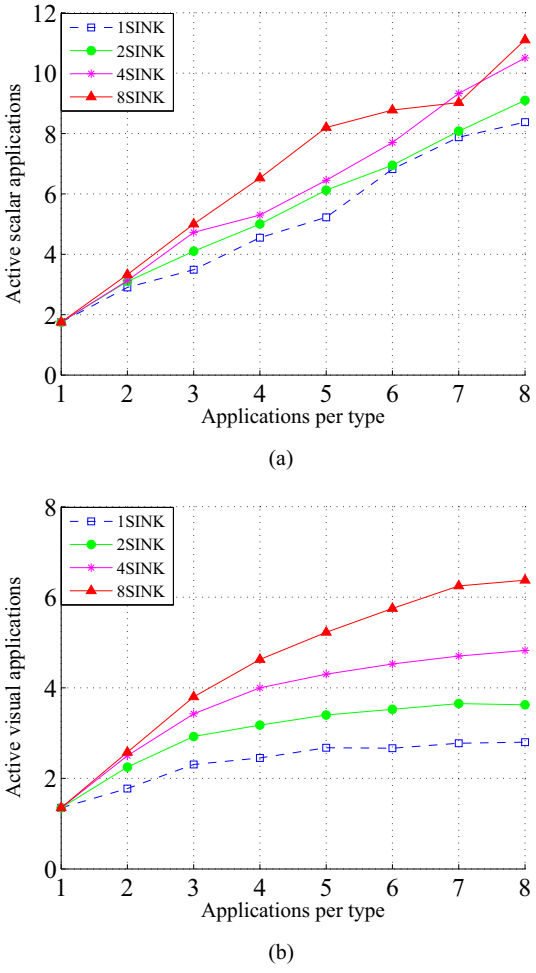


Fig. 7. Number of active applications vs offered applications per type varying the number of sinks. (a) Scalar (b) Visual.

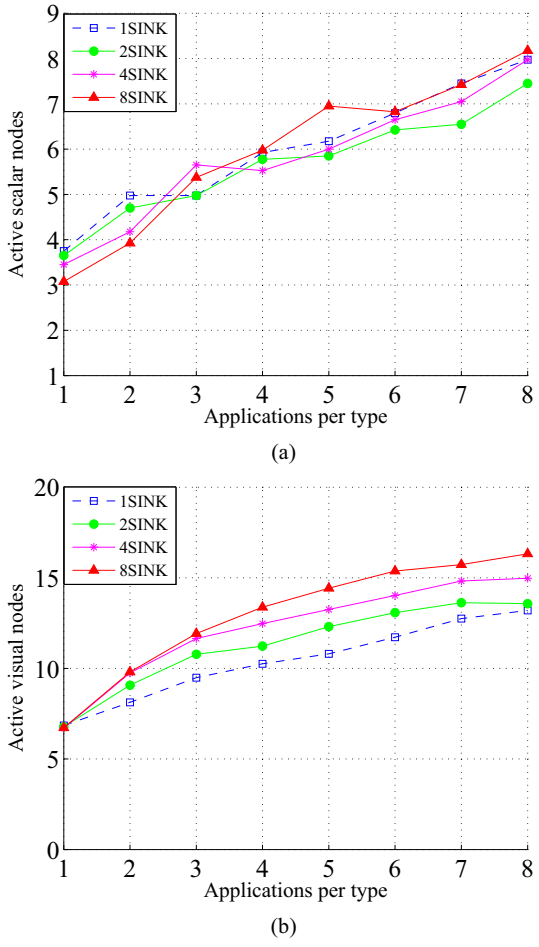


Fig. 8. Number of active nodes vs offered applications per type varying the number of sinks. (a) Scalar (b) Visual.

5.4.3. Lifetime

In the reference scenario, the minimum lifetime of the virtual sensor network is set to $L = 1$ day. It is worth evaluating the impact of L on the performance of the virtualization framework. To this extent, we run experiments varying L from 1 hour to 8 days. The corresponding cumulative results are reported in Table 6, whereas the detailed ones are given in Figs. 9 and 10. Again, the virtualization approach allows to achieve considerable gains (in terms of deployed applications) for all the tested values of the lifetime constraints.

Please cite this article as: C. Delgado et al., On optimal resource allocation in virtual sensor networks, Ad Hoc Networks (2016), http://dx.doi.org/10.1016/j.adhoc.2016.04.004

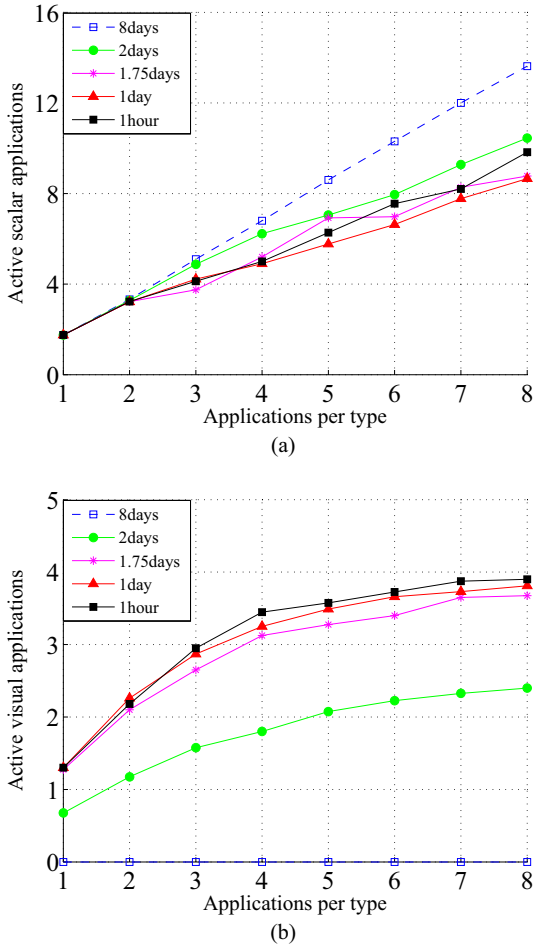


Fig. 9. Number of active applications vs offered applications per type varying the network lifetime. (a) Scalar (b) Visual.

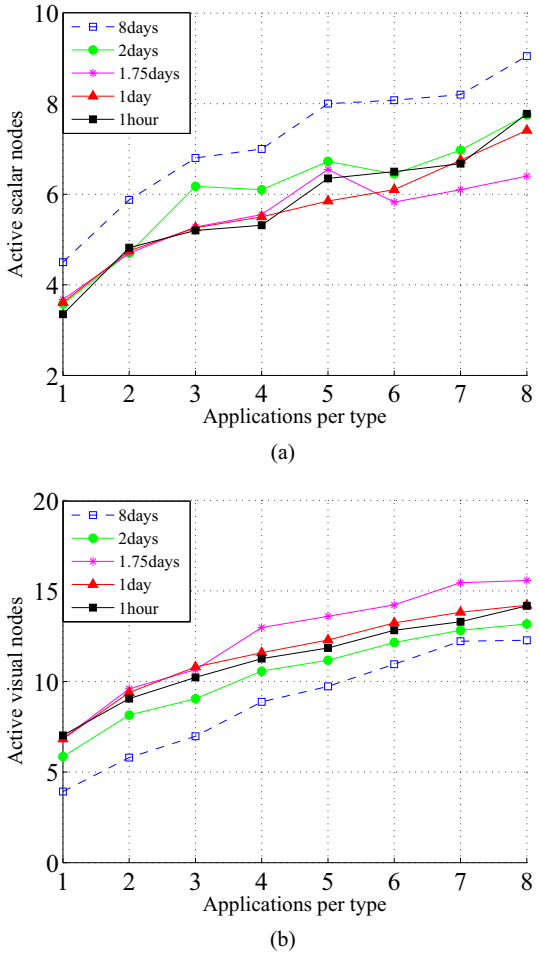


Fig. 10. Number of active nodes vs offered applications per type varying the network lifetime. (a) Scalar (b) Visual.

Fig. 9 (b) further shows that with $L = 8$ days, visual applications cannot be activated since multimedia motes do not have enough energy to support any visual application. Logically, multimedia motes (Fig. 10(b)) are still activated because they can be used by the scalar applications. In addition, there is a remarkable decrease in the number of active visual applications from $L = 1.75$ to $L = 2$ days. The reason is that ATC visual applications, which demand more energy, cannot be activated with $L = 2$ days and the only active visual applications are the CTA ones. It is also interesting to observe that from $L = 1$ to $L = 1.75$ days there is a slight decrease in the number of active visual applications whereas the number of active multimedia motes rises. This is due to the fact that nodes that were simultaneously sensing several test points when $L = 1$, do not have now enough energy when $L = 1.75$ and therefore, additional nodes must be activated.

5.4.4. Type of routing

Finally, the three different types of routing described in Section 3.2 are analyzed: *multipath*, *singlepath* and *static routing*. Table 7 reports the total number of deployed applications and the corresponding value of the optimization objective function (1) under different routing paradigms when considering a maximum emitted power of $P_{max} = -10$ dBm (maximum transmission range of 33 m).

It can be observed that the *singlepath* routing achieves a performance very close to the upper bound provided by the ideal non-constrained *multipath* routing. In addition, the much simpler *static* routing is also close to the *singlepath*. Figs. 11 and 12 show the breakdown of the number of active applications and the number of active nodes for the three different routing schemes in the same setting of Table 7. It can be further noted that the number of active nodes (Fig. 12) rises due to the increase in the number of hops (since maximum transmission power has been reduced in this section).

5.5. Performance of heuristic algorithm

In this section, we evaluate the performance of the heuristic algorithm in terms of optimality gap and solution time.

To this purpose, we increase the size of the reference network topology while maintaining the same node density and a maximum emitted power of $P_{max} = -10$ dBm. Table 8 summarizes the main features of the tested scenarios. For example, the first one is a 200 × 200 m scenario with 36 TelosB motes (one of them acting as a sink), 36 BeagleBone (also one of them acting as a sink) and six applications of each type (six light monitoring, six temperature monitoring, six ATC and six CTA). The following results have been obtained considering *static* routing. Fig. 13 reports the optimality gap and the solution time of the heuristic; the results have been obtained on 3.0 Ghz Quad Core Intel Woodcrest (64bits) machines with 8Gb RAM and 250Gb SATA storage, averaging over 100 randomly generated network topologies for each scenario of Table 8. These results show that the proposed heuristic approach has a limited optimality gap with respect to the optimal solution while featuring low computation time even for “large” network topologies.

Please cite this article as: C. Delgado et al., On optimal resource allocation in virtual sensor networks, Ad Hoc Networks (2016), http://dx.doi.org/10.1016/j.adhoc.2016.04.004

Table 7
Total number of deployed applications and value of the optimization objective function (1) when varying the reference routing paradigm. Column (a) reports the reference routing paradigm; column (b) reports the number of "offered" applications.

		SEPARATE		JOINT	
(a)	(b)	Total applications	Objective function	Total applications	Objective function
Static	2	4.03	13.58	5.65	26.07
	4	7.05	19.49	8.23	34.63
	6	9.38	22.79	10.48	38.73
Singlepath	2	4.31	17.36	6.03	28.31
	4	7.43	23.62	8.89	35.19
	6	9.89	28.12	11.00	39.40
Multipath	2	4.32	17.48	5.87	28.86
	4	7.47	23.97	8.77	36.99
	6	9.91	28.44	11.08	40.56

Table 8
Scenario topologies.

	Scenario			
	1	2	3	4
Size	200 × 200 m	283 × 283 m	346 × 346 m	400 × 400 m
Number of nodes	36 + 36	72 + 72	108 + 108	144 + 144
Number of sinks	1 + 1	2 + 2	3 + 3	4 + 4
Number of applications	6+6+6+6	12+12+12+12	18+18+18+18	24+24+24+24

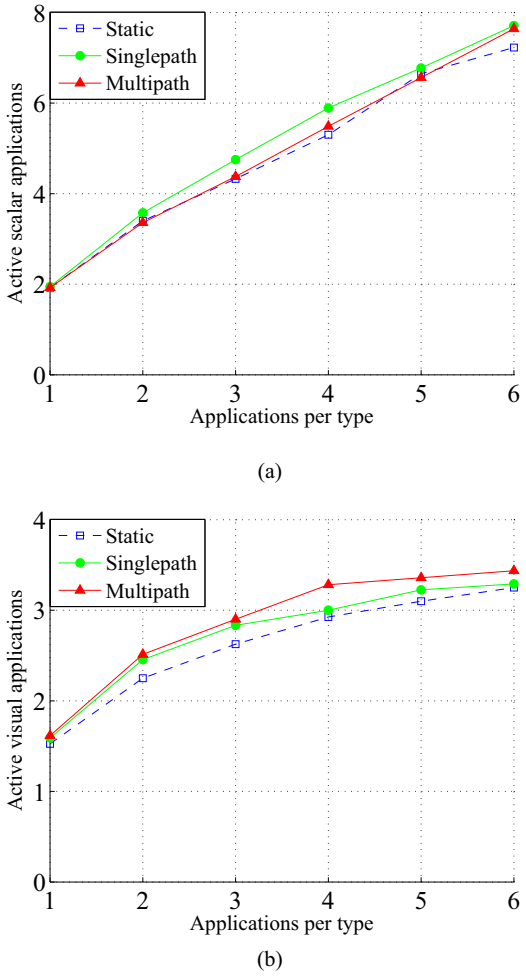


Fig. 11. Number of active applications vs offered applications per type varying the routing schemes. $P_{max} = -10\,dBm$. (a) Scalar (b) Visual.

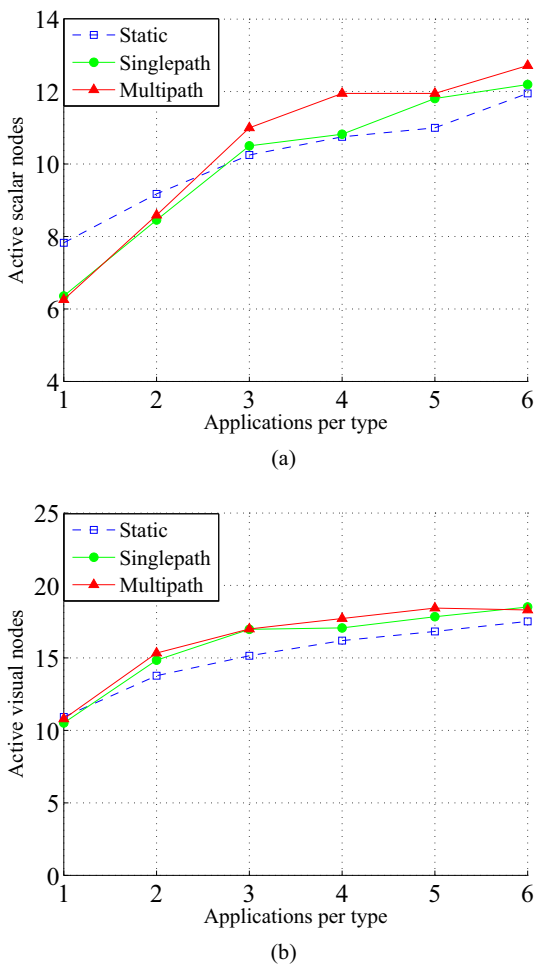
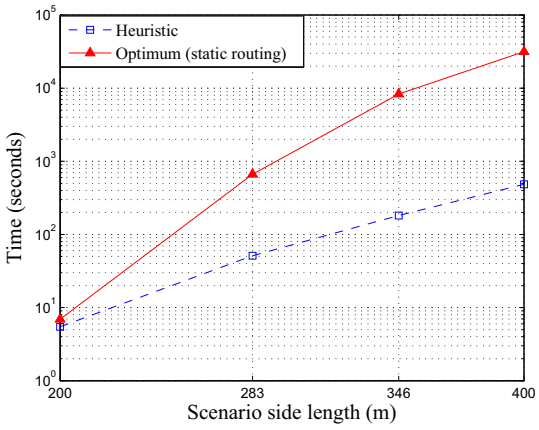
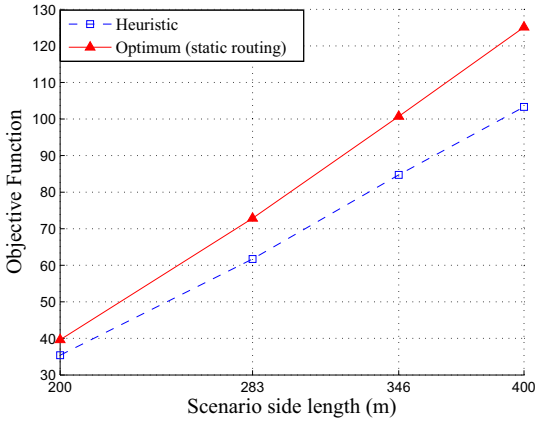


Fig. 12. Number of active nodes vs offered applications per type varying the routing schemes. $P_{max} = -10\,dBm$. (a) Scalar (b) Visual.

Please cite this article as: C. Delgado et al., On optimal resource allocation in virtual sensor networks, Ad Hoc Networks (2016), http://dx.doi.org/10.1016/j.adhoc.2016.04.004



(a)



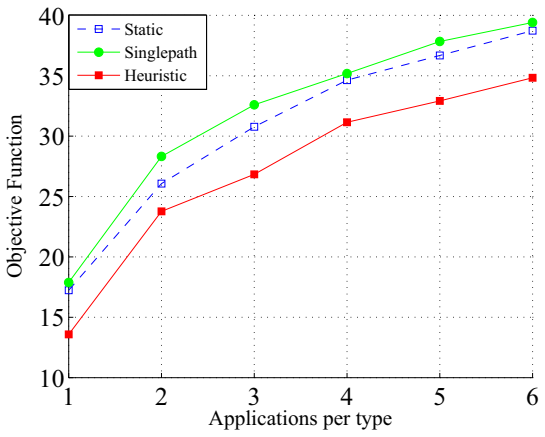
(b)

Fig. 13. Performance evaluation of the heuristic algorithm vs the optimum scheme with *static* routing. (a) Computation Time (in log scale) (b) Objective function.

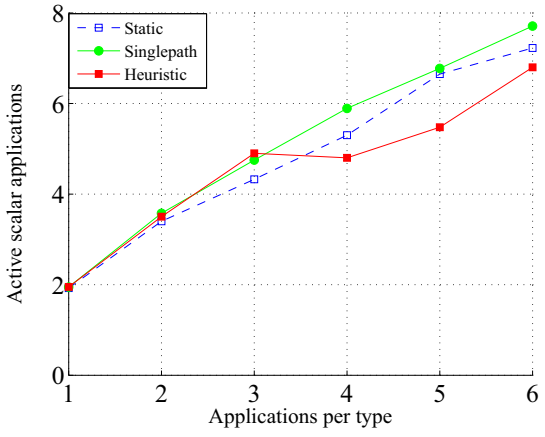
Fig. 14 gives more insight on the the performance of the proposed heuristic algorithm. Here it is compared with the optimum solutions achieved with the *singlepath* routing and with the *static* routing in the 200 × 200 m scenario when varying the number of offered applications (note that results shown for six offered applications corresponds to the scenario 1 in Fig 13). The breakdown of the deployed applications additionally shows a similar degradation in the number of active scalar and visual applications (Figs. 14(b) and 14(c)). All these results suggest the potential of this algorithm as a centralized resource allocation tool for virtual sensor networks.

6. Conclusion

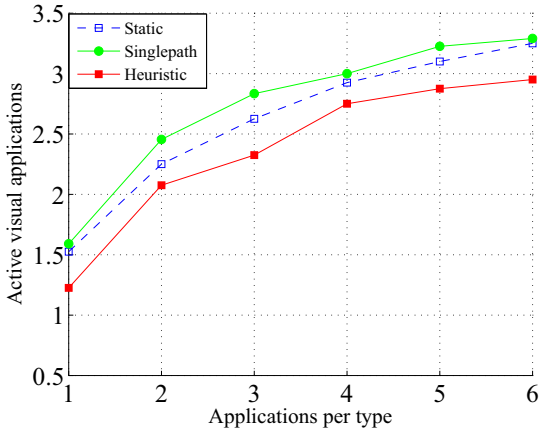
In this paper we have analyzed the benefit of virtualization in the field of wireless sensor networks. Namely, we have proposed an optimization approach to assign the resources of a shared sensor network to multiple different applications. The proposed approach targets the maximization of the overall revenue out of the application deployment process (that is, number of deployed applications) while minimizing the cost of the required physical infrastructure (number of sensor nodes). The proposed approach accounts for constraints on the sensor nodes capabilities (memory, computation, energy) and network limitations (topology, shared bandwidth). Given the complexity of the proposed optimization problem, a heuristic algorithm is also proposed to obtain suboptimal solutions in limited computation time.



(a)



(b)



(c)

Fig. 14. Performance of heuristic algorithm vs the optimum scheme with *singlepath* and *static* routing. Number of active applications vs offered applications per type. (a) Objective function (b) Active scalar applications (c) Active visual applications.

The proposed optimization framework and the heuristic algorithm have been leveraged to assess the gain of sharing a general purpose wireless sensor network with respect to “vertical” network deployments in which several physical infrastructures, each one optimized on the requirements of a single application, coexist. The results obtained for realistic network scenarios show that optimally orchestrating a shared infrastructure provides better performance with respect to stand-alone vertical network deployments

Please cite this article as: C. Delgado et al., On optimal resource allocation in virtual sensor networks, Ad Hoc Networks (2016), http://dx.doi.org/10.1016/j.adhoc.2016.04.004

(in terms of total number of deployable applications and total cost of the used physical infrastructure).

Acknowledgment

This work has been supported by the Spanish Government through the grant TEC2014-52969-R from the Ministerio de Ciencia e Innovación (MICINN), Gobierno de Aragón (research group T98), the European Social Fund (ESF), Centro Universitario de la Defensa through project CUD2013-05, Universidad de Zaragoza, Fundación Bancaria Ibercaja and Fundación CAI (IT 2/15).

This work has also been partially supported by the Italian Ministry for Education, University and Research (MIUR) through the national cluster project SHELL, Smart Living technologies (grant number: CTN01 00128 111357).

Appendix A. Visual applications parameters

Referring to [42], it is assumed that different techniques for the extraction of local visual features are used: CTA will use the SIFT (Scale Invariant Feature Transform) algorithm while ATC will use BRISK (Binary Robust Invariant Scalable Keypoints) algorithm. Assuming a desired Mean of Average Precision (MAP) of 0.6, the use of Zurich Building Database (ZuBuD) [43] and an application frame rate of $\lambda = 1$ query per second for both CTA and ATC paradigms, the required capacity will be 20 kb/s for CTA-SIFT and 12 kb/s for ATC-BRISK [42].

For this kind of applications, the energy consumed to process the data is not negligible. In [42] a characterization of this energy on a BeagleBone-based visual sensor node is provided. The processing energy for the CTA paradigm can be computed as:

$$E_{cpu}^{CTA}(\rho) = P_{cpu} \cdot t_{cpu}^{CTA}(\rho) \tag{A.1}$$

where P_{cpu} is the power dissipated by the processor of the visual sensor node and has a value of 2.1 W for BeagleBone sensor nodes; and $t_{cpu}^{CTA}(\rho)$ is the time required to process an image, which depends on ρ, the amount of sent information per query (20 kbs in our scenario). According to the results in [44], the processing energy for an image for the CTA application in our scenario is 0.05 J. Therefore, assuming a frame rate of $\lambda = 1$ query per second, the power dissipation (function f in Eq. (19)) is 0.05 W. In addition, we can estimate the required processing load l_j for a BeagleBone as the fraction of time used by the application ($t_{cpu}^{CTA} \cdot \lambda$) multiplied by the processing power of the sensor node, L_i. In this case, $24.5 \cdot 1 \cdot 720 = 17.64$ MIPS.

Similarly, the processing energy for the ATC paradigm can be computed as:

$$E_{cpu}^{ATC}(\rho) = P_{cpu} \cdot \left[\tau_{off} + M(\rho) \cdot (\tau_{det} + \tau_{desc})\right] \tag{A.2}$$

where τ_{off} is the time spent for initializing the detector and has a value of $1.6 \cdot 10^{-4}$ ms/pixel. With an image size of 640 × 480 pixels, τ_{off} is 49.152 ms. τ_{det} and τ_{desc} are the time spent for detecting and describing one BRISK feature of the image and their values are 0.31 ms and 0.16 ms respectively. $M(\rho)$ is the optimal number of features that depends on the rate ρ. For $\rho = 12$ kb/query, the minimum value of M to provide a MAP of 0.6 is $M = 100$ features. Thus the processing energy for an image for the ATC application in our scenario is 0.2 J, and the power dissipation is 0.2 W. The processing load in this case is 69.23 MIPS.

Regarding memory requirements, specific values have not been obtained for these applications. However, given the great difference in the amount of available memory in TelosB (10 KB) and BeagleBone (256 MB), we are assuming that due to memory constraints, multimedia applications could not be implemented in TelosB nodes and memory will not be a limiting factor in BeagleBone nodes, since processing or transmission rate will limit long before these applications rather than memory. Summing up, the requirements vector for CTA and ATC applications are respectively $r_j = \{20 \text{ kb/s}, 10 \text{ KB} < m_j << 256 \text{ MB}, 17.64 \text{ MIPS}\} r_j = \{12 \text{ kb/s}, 10\text{KB} < m_j << 256 \text{ MB}, 69.23 \text{ MIPS}\}$.

References

[1] A. Jayasumana, H. Qi, T. Illangasekare, Virtual sensor networks - a resource efficient approach for concurrent applications, in: Proceedings 4th International Conference on Information Technology (ITNG'07), Las Vegas, 2007, pp. 111–115.
[2] M.M. Islam, E.-N. Huh, Virtualization in wireless sensor network: challenges and opportunities, J. Netw. 7 (3) (2012) 412–418.
[3] T. Luo, H.-P. Tan, T. Quek, Sensor openflow: enabling software-defined wireless sensor networks, Commun. Lett. IEEE 16 (11) (2012) 1896–1899, doi:10.1109/LCOMM.2012.092812.121712.
[4] L. Sarakis, T. Zahariadis, H.-C. Leligou, M. Dohler, A framework for service provisioning in virtual sensor networks, EURASIP J. Wirel. Commun. Netw. (2012).
[5] A. Merentitis, F. Zeiger, M. Huber, N. Frangiadakis, K. Mathioudakis, K. Sasloglo, G. Mazarakis, V. Gazis, Z. Boufidis, WSN trends: sensor infrastructure virtualization as a driver towards the evolution of the internet of things, in: Proceedings of the Seventh International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies UBICOMM 2013, 2013, pp. 113–118.
[6] M.K. Chowdhury, R. Boutaba, A survey of network virtualization, Comput. Netw. 54 (5) (2010) 862–876.
[7] Z. Xiao, W. Song, Q. Chen, Dynamic resource allocation using virtual machines for cloud computing environment, IEEE Trans. Parallel Distr. Syst. 24 (6) (2013) 1107–1117.
[8] A. Fischer, J. Botero, M.T. Beck, H. de Meer, Virtual network embedding: a survey, IEEE Commun. Surv. Tut. 15 (4) (2013) 1888–1906.
[9] M.M. Islam, M.M. Hassan, G.-W. Lee, E.-N. Huh, A survey on virtualization of wireless sensor networks, Sensors 12 (2012) 2175–2207.
[10] S. Madria, V. Kumar, R. Dalvi, Sensor cloud: a cloud of virtual sensors, IEEE Softw. 31 (2) (2014) 70–77, doi:10.1109/MS.2013.141.
[11] I. Khan, F. Belqasmi, R. Glitho, N. Crespi, M. Morrow, P. Polakos, Wireless sensor network virtualization: early architecture and research perspectives, IEEE Network 29 (2015) 104–112.
[12] P. Levis, D. Culler, Mate: a tiny virtual machine for sensor networks, SIGARCH Comput. Archit. News 30 (5) (2002) 85–95, doi:10.1145/635506.605407.
[13] P. Levis, D. Gay, D. Culler, Active sensor networks, in: Proceedings of the 2nd Conference on Symposium on Networked Systems Design & Implementation, in: NSDI'05, vol. 2, USENIX Association, Berkeley, CA, USA, 2005, pp. 343–356. http://dl.acm.org/citation.cfm?id=1251203.1251228
[14] Y. Yu, L.J. Rittle, V. Bhandari, J.B. LeBrun, Supporting concurrent applications in wireless sensor networks, in: Proceedings of the 4th International Conference on Embedded Networked Sensor Systems, in: SenSys '06, ACM, New York, NY, USA, 2006, pp. 139–152, doi:10.1145/1182807.1182822.
[15] J. Koshy, R. Pandey, Vmstar: synthesizing scalable runtime environments for sensor networks, in: Proceedings of the 3rd International Conference on Embedded Networked Sensor Systems, in: SenSys '05, ACM, New York, NY, USA, 2005, pp. 243–254, doi:10.1145/1098918.1098945.
[16] I. Leontiadis, C. Efstratiou, C. Mascolo, J. Crowcroft, Senshare: transforming sensor networks into multi-application sensing infrastructures, in: Proceedings of the 9th European Conference on Wireless Sensor Networks, in: EWSN'12, 2012, pp. 65–81.
[17] S. Bhattacharya, A. Saifullah, C. Lu, G. Roman, Multi-application deployment in shared sensor networks based on quality of monitoring, in: Real-Time and Embedded Technology and Applications Symposium (RTAS), 2010 16th IEEE, 2010, pp. 259–268, doi:10.1109/RTAS.2010.20.
[18] C.-L. Fok, C. Julien, G.-C. Roman, C. Lu, Challenges of satisfying multiple stakeholders: Quality of service in the internet of things, in: Proceedings of the 2nd Workshop on Software Engineering for Sensor Network Applications, in: SESENA '11, ACM, New York, NY, USA, 2011, pp. 55–60, doi:10.1145/1988051.1988062.
[19] W. Li, F.C. Delicato, P.F. Pires, Y.C. Lee, A.Y. Zomaya, C. Miceli, L. Pirmez, Efficient allocation of resources in multiple heterogeneous wireless sensor networks, J. Parallel Distrib Comput 74 (1) (2014) 1775–1788. http://www.sciencedirect.com/science/article/pii/S0743731513002104
[20] R. Huang, X. Chu, J. Zhang, Y.H. Hu, Energy-efficient monitoring in software defined wireless sensor networks using reinforcement learning: A prototype, Int J Distrib Sensor Netw 2015 (2015), doi:10.1155/2015/360428.
[21] Y. Xu, A. Saifullah, Y. Chen, C. Lu, S. Bhattacharya, Near optimal multi-application allocation in shared sensor networks, in: Proceedings of the Eleventh ACM International Symposium on Mobile Ad Hoc Networking and Computing, in: MobiHoc '10, ACM, New York, NY, USA, 2010, pp. 181–190, doi:10.1145/1860093.1860118.
[22] C. Wu, Y. Xu, Y. Chen, C. Lu, Submodular game for distributed application allocation in shared sensor networks, in: INFOCOM, 2012 Proceedings IEEE, 2012, pp. 127–135, doi:10.1109/INFCOM.2012.6195490.
[23] C. de Farias, L. Pirmez, F. Delicato, W. Li, A. Zomaya, J. De Souza, A scheduling algorithm for shared sensor and actuator networks, in: Information Networking (ICOIN), 2013 International Conference on, 2013, pp. 648–653, doi:10.1109/ICOIN.2013.6496703.

[24] D. Zeng, P. Li, S. Guo, T. Miyazaki, J. Hu, Y. Xiang, Energy minimization in multi-task software-defined sensor networks, IEEE Trans. Comput. 64 (11) (2015) 3128–3139, doi:10.1109/TC.2015.2389802.
[25] N. Edalat, W. Xiao, M. Motani, N. Roy, S.K. Das, Auction-based task allocation with trust management for shared sensor networks, Secur. Commun. Netw. 5 (11) (2012) 1223–1234, doi:10.1002/sec.631.
[26] C. Delgado, J.R. Gállego, M. Canales, J. Ortín, S. Bousnina, M. Cesana, An optimization framework for resource allocation in virtual sensor networks, in: Proceedings of IEEE Global Communications Conference, in: Globecom '15, IEEE, 2015.
[27] Y. Shi, T. Hou, J. Liu, S. Kompella, Bridging the gap between protocol and physical models for wireless networks, IEEE Trans. Mobile Comput. 12 (7) (2013) 1404–1416.
[28] T. Winter, P. Thubert, A. Brandt, J. Hui, R. Kelsey, P. Levis, K. Pister, R. Struik, J. Vasseur, R. Alexander, RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks, 2012, (RFC 6550 (Proposed Standard)).
[29] I. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, Wireless sensor networks: a survey, Comput. Netw. 38 (4) (2002) 393–422.
[30] A. Redondi, M. Tagliasacchi, M. Cesana, Rate-accuracy optimization in visual wireless sensor networks, in: IEEE International Conference on Image Processing (ICIP2012), Orlando, Florida, 2012, pp. 1105–1108.
[31] T. Hou, Y. Shi, H. Sherali, Rate allocation and network lifetime problems for wireless sensor networks, IEEE/ACM Trans. Netw. 16 (2) (2008) 321–334.
[32] M.R. Garey, D.S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, W. H. Freeman & Co., New York, NY, USA, 1990.
[33] W. Li, S. Wang, Y. Cui, X. Cheng, R. Xin, M.A. Al-Rodhaan, A. Al-Dhelaan, AP association for proportional fairness in multirate WLANs, IEEE/ACM Trans. Netw. 22 (1) (2014) 191–202, doi:10.1109/TNET.2013.2245145.
[34] L. Li, M. Pal, Y. Yang, Proportional fairness in multi-rate wireless LANs, in: Proceedings IEEE INFOCOM 2008, Phoeniz, AZ, 2008, pp. 1678–1686.
[35] ILOG CPLEX, 2015, (http://www-01.ibm.com/software/integration/optimization/cplex-optimizer/).
[36] A. Chen, S. Kumar, T.H. Lai, Designing localized algorithms for barrier coverage, in: Proceedings of the 13th Annual ACM International Conference on Mobile Computing and Networking, in: MobiCom '07, ACM, New York, NY, USA, 2007, pp. 63–74.
[37] C. Suh, J.-E. Joung, Y.-B. Ko, New RF models of the tinyos simulator for IEEE 802.15.4 standard, in: Proceedings IEEE WCNC 2007, Hong Kong, 2007, pp. 2238–2242.
[38] CC2420: "Datasheet for Chipcon (TI) CC2420 2.4 GHz IEEE 802.15.4/ZigBee RF Transceiver".
[39] TelosB Mote Platform Datasheet, MEMSIC Inc.
[40] G. Coley, Beaglebone REV A6 system reference manual, 2012.
[41] J.B. Javier Molina, Javier M. Mora-merchan, C. Leon, Wireless Sensor Networks: Application, InTech. Doi: 10.5772/13398.
[42] A. Redondi, L. Baroffio, L. Bianchi, M. Cesana, M. Tagliasacchi, Compress-then-analyze vs analyze-then-compress: what is best in visual sensor networks? IEEE Trans. Mobile Comput. PP (99) (2016), doi:10.1109/TMC.2016.2519340. 1–1.
[43] H. Shao, T. Svoboda, L.V. Gool, Zubud zurich buildings database for image based recognition, Tech. Rep. 260 (2003). Computer Vision Laboratory, Swiss Federal Institute of Technology.
[44] A.E. Redondi, Energy-aware visual analysis for wireless multimedia sensor networks, Politecnico Di Milano, 2014 Ph.D. thesis.

Please cite this article as: C. Delgado et al., On optimal resource allocation in virtual sensor networks, Ad Hoc Networks (2016), http://dx.doi.org/10.1016/j.adhoc.2016.04.004

Carmen Delgado was born in Logroño (Spain) in 1989. She received the engineer of telecommunications MS and the biomedical engineering MS degrees from the Universidad de Zaragoza, Spain, in 2013 and 2014, respectively. She is currently pursuing the Ph.D. degree in mobile network information and communication technologies with the Departamento de Ingeniería Electrónica y Comunicaciones, Universidad de Zaragoza, Spain. Her research interest lies in the field of wireless sensor networks.

José Ramón Gállego was born in Zaragoza (Spain) on 1978. He received the engineer of telecommunications MS and Ph.D. degrees from the Universidad de Zaragoza, Spain, in 2001 and 2007, respectively. In 2002, he joined the Departamento de Ingeniería Electrónica y Comunicaciones, Universidad de Zaragoza, where he is currently an Associate Professor. He is member of the Aragón Institute of Engineering Research (I3A). His professional research activity lies in the field of wireless communications, with emphasis on radio resource management wireless sensor networks and cognitive networks.

María Canales was born in Zaragoza (Spain) on 1978. She received the engineer of telecommunications MS and Ph.D. degrees from the Universidad de Zaragoza, Spain, in 2001 and 2007, respectively. In 2002, she joined the Departamento de Ingeniería Electrónica y Comunicaciones, Universidad de Zaragoza, where she is currently an associate professor. She is member of the Aragón Institute of Engineering Research (I3A). Her professional research activity lies in the field of wireless communications, with emphasis on radio resource management wireless sensor networks and cognitive networks.

Jorge Ortín was born in Zaragoza, Spain, in 1981. He received the engineer of telecommunications and Ph.D. degrees from the Universidad de Zaragoza in 2005 and 2011 respectively. In 2008 he joined Aragón Institute of Engineering Research (I3A) of Universidad de Zaragoza, where he has participated in different projects funded by public administrations and by major industrial and mobile companies. In 2012 he joined the Universidad Carlos III of Madrid as a postdoc research fellow. From 2013, he works at Centro Universitario de la Defensa Zaragoza. Research interests include wireless communications systems.

Sonda Bousnina was born in Sfax, Tunisia, in 1986. She received the engineering degree in computer science, professional master degree in security of information systems and the research masters degree in new technologies of dedicated computer systems from the National School of Engineers of Sfax (ENIS), Tunisia, in June 2009, October 2009 and March 2012, respectively. Then, she started the Ph.D. studies at REGIM Laboratory (Research Groups on Intelligent Machines) in ENIS, Sfax. In November 2013, she obtained a scholarship in order to continue her Ph.D. studies at Polytechnic of Milan and she is currently a Ph.D. Student in telecommunication at ANT Lab (Advanced Network Technologies Laboratory).

Matteo Cesana is currently an associate professor with the Dipartimento di Elettronica, Informazione e Bioingegneria of the Politecnico di Milano, Italy. He received his MS degree in telecommunications engineering and his Ph.D. degree in information engineering from Politecnico di Milano in July 2000 and in September 2004, respectively. From September 2002 to March 2003 he was a visiting researcher at the Computer Science Department of the University of California in Los Angeles (UCLA). His research activities are in the field of design, optimization and performance evaluation of wireless networks with a specific focus on wireless sensor networks and cognitive radio networks. Dr. Cesana is an associate editor of the *Ad Hoc Networks Journal* (Elsevier).

Please cite this article as: C. Delgado et al., On optimal resource allocation in virtual sensor networks, Ad Hoc Networks (2016), http://dx.doi.org/10.1016/j.adhoc.2016.04.004