Ad Hoc Networks 11 (2013) 712–732

Contents lists available at ScienceDirect

Ad Hoc Networks

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

Cross-layer design and analysis of WSN-based mobile target detection systems

Paolo Medagliani [a,1], Gianluigi Ferrari [b,*], Vincent Gay [c], Jérémie Leguay [c]

[a] *Lepida spa, Viale Aldo Moro 64, I-40127 Bologna, Italy*
[b] *WASN Lab, Dept. of Inf. Engin., Univ. of Parma, Viale G.P. Usberti 181/A, I-43124 Parma, Italy*
[c] *Thales Communications, 160 Bd de Valmy, Colombes Cedex, France*

ARTICLE INFO

Article history:
Received 15 October 2010
Received in revised form 2 April 2011
Accepted 12 July 2011
Available online 27 July 2011

Keywords:
Wireless Sensor Network (WSN)
Cross-layer energy model
Probability of detection
Latency of communication
X-MAC
Cas-MAC protocol

ABSTRACT

The limited node capabilities typical of Wireless Sensor Networks (WSNs) call for cross-layer design optimization. In this paper, we address the problem of designing and operating long-lasting surveillance mobile target detection applications for unattended WSNs with a priori knowledge of the nodes' positions. In particular, we focus on the cross-layer interaction between the sensing layer (devoted to the detection of a mobile target crossing the monitored area) and the communication layer (devoted to the transmission of the alert, upon detection, from a sensing node to the network sink). The performance of the sensing layer is characterized by the probability of target missed detection and the delay before the first sensor detection act. The communication layer is investigated considering two Medium Access Control (MAC) protocols: X-MAC [1] and the novel Cascade (Cas)-MAC protocol, inspired by the principles of the D-MAC protocol [2]. At both layers, we validate analytical models through realistic simulations and experiments. The cross-layer interaction between the two layers is achieved considering a proper model for the network lifetime, based on the average energy depletion at the node level. Finally, to highlight the benefits of the proposed framework, we present a cross-layer optimization approach for the configuration of the system parameters, considering several relevant network topologies.

1. Introduction

Wireless Sensor Networks (WSNs) are commonly used for environmental monitoring, surveillance operations, and home or industrial automation. These networks are typically composed of small form factor sensors (or actuators) that have limited resources in terms of processing power, data storage, and radio transmission [3]. Most of the time, WSN nodes operate on batteries and can monitor simple phenomena as they embed physical transducers with basic processing capabilities. For instance, in military scenarios, seismic and magnetic sensors can be left unattended to detect motorized vehicles in a given area [4]. Recent advances in hardware miniaturization, low-power radio communications, and battery lifetime, together with the increasing affordability of such devices, are paving the road for a widespread use of WSNs.

Through the use of embedded transducers, such as acoustic, seismic or infrared sensors, WSN nodes can perform local or collaborative target signature detection and can consequently trigger actuators (e.g., flash lights, sirens). Despite their limited individual detection capabilities, a large number of networked sensors can lead to powerful WSN-based surveillance systems [5]. WSN nodes can be easily deployed and recovered, are lightweight, and provide cost-effective complements to existing surveillance systems (e.g., cameras, radars). They can help securing

[*] Corresponding author. Tel.: +39 0521 906513; fax: +39 0521 905758.
E-mail addresses: paolo.medagliani@lepida.it (P. Medagliani), gianluigi.ferrari@unipr.it (G. Ferrari), vincent.GAY@fr.thalesgroup.com (V. Gay), jeremie.LEGUAY@fr.thalesgroup.com (J. Leguay).
[1] The largest part of this work was carried out while Paolo Medagliani was with the University of Parma.

1570-8705/$ - see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.adhoc.2011.07.009

and protecting people and assets in remote or inaccessible areas.

This paper focuses, without loss of generality, on surveillance applications where WSNs perform continuous and long lasting detection of mobile targets (e.g., pedestrians, vehicles) in large areas. In this case, WSN nodes are often referred to as Unattended Ground Sensors (UGSs) [4]. In such vast and long-term deployments using resource-constrained devices, one of the main design goals is to maximize the operational lifetime of the system while ensuring high target detection performance and short response times. It will be shown that system optimization calls for a cross-layer interaction between sensing (for detection) and communication layers. Indeed, a number of system parameters and functioning modes contribute to a general trade-off between the energy consumption and the quality of service (in terms of detection capabilities and system response time). Therefore, the configuration and the design of WSN-based detection systems is complicated and represents a challenging cross-layer issue.

This paper proposes a cross-layer model, which can be used as an engineering toolbox, to efficiently deploy a WSN. This cross-layer model analyzes the performance of key functions such as sensing and communication, which correspond to two critical system layers. Since the same battery is used by a node for detection and communication purposes, it follows that the node energy model is a relevant cross-layer system aspect. More precisely, we consider battery-powered nodes that cyclically switch on and off their sensing and communication modules to save energy. By tuning these duty cycles, the system lifetime can be extended at the price of lower detection capability and system reactivity. In addition, we assume that the node placement is a priori known: this is accurate for WSN-based surveillance systems, as the nodes are deployed by the network operator. Knowledge of the exact network topology is used to bound, for a given mobile target model, the target detection probability. It is also shown that node placement has an impact on the alert transmission delay, as it determines the number of hops from a sensor node to the sink.

In [6], the performance of WSN-based detection systems with *stochastic* node placement is analyzed. This allows to evaluate the *average* system performance, without, however, evidencing the cross-layer impact of the WSN topology.

The current paper extends prior work in [6,7], by considering deterministic (and, thus, a priori known) node placement and a larger number of performance indicators. This allows to evaluate more accurately and more comprehensively the performance of WSN-based target detection systems, notably taking into account the specific topology of the WSN. The system performance is analyzed and assessed through realistic simulations and experiments, considering a variety of performance indicators, including the probability of target missed detection, the delay before first detection, the latency after detection, and the network lifetime. The latter indicators are described hereafter, as well as the optimization techniques, pointing out the significant additional contributions of this paper with respect to previous work.

- *Probability of target missed detection (sensing layer)*: this indicator denotes the detection capability of a WSN. In [7], we extend the statistical average evaluation proposed in [6], deriving lower and upper bounds which depend on the specific topology at hand. In this paper, we extend previous work by refining the derivation of lower and upper bounds and evaluating more extensively their accuracy.
- *Delay before first detection (sensing layer)*: this newly introduced performance indicator corresponds to the time interval between the instant at which a target enters the monitored area and the instant of first detection act by a WSN node.
- *Latency after detection (communication layer)*: this indicator represents the time interval between target detection at a WSN node and alert reception at the sink. This indicator is representative of the network reactivity to an intrusion detection and mainly depends on the Medium Access Control (MAC) protocol. The latency is evaluated considering, in a comparative way, the X-MAC protocol [1], already considered in [6] in the presence of stochastic deployment, with a novel MAC protocol for surveillance applications, inspired by the principles of the D-MAC protocol [2] and denoted as Cascade (Cas) MAC, whose design aims at efficiently supporting the dominant traffic pattern—namely, infrequent alerts sent to the sink under tight latency constraints. The Cas-MAC protocol minimizes the latency after detection while nodes operate in very low-power modes.
- *Network lifetime (cross-layer)*: the energy of a WSN node is depleted by both sensing and communication operations. As such, it has a direct impact on the network lifetime, defined as the duration during which the WSN operates before the average residual energy of at least one of the nodes in the network reduces to zero. In this paper, we extend the derivation of the model for the average network lifetime, already considered in [6] with the use of the X-MAC protocol, by considering the specific design of the newly introduced Cas-MAC protocol.
- *Optimization toolkit (cross-layer)*: we extend the optimization toolkit introduced in [6] to take into account deterministic node placement. This allows to directly understand the cross-layer impact of the network topology.

At the sensing layer, the probability of target missed detection will be investigated as a function of the number of sensing nodes and of the sensing duty cycle. It will be shown that the probability of target missed detection can be increased at the cost of longer delay before first detection. At the communication layer, it will be shown that the Cas-MAC protocol allows to reach better cross-layer trade-offs, in terms of network lifetime and latency, especially in very low-power modes.

This paper is structured as follows. Section 2 is dedicated to related works. In Section 3, accurate bounds for the probability of target missed detection are derived. In Section 4, the average delay before the first sensor detection is computed. Section 5 is devoted to the evaluation of the latency after detection, considering X-MAC and Cas-MAC protocols. In Section 6, a cross-layer node energy model is presented. In Section 7, cross-layer optimization of a few representative WSN-based target detection systems is considered. Section 8 concludes the paper.

2. Related work

In the literature, the cross-layer design of WSNs has received a significant attention in recent years. In [8,9], the authors review the state-of-art techniques and propose a taxonomy including a power efficient approach similar to that considered in the remainder of the current paper through the use of duty cycling. They also underline the need for analytical tools able to predict the system performance. In [10–12], the authors present ANDES, an analytical design tool, which allows to quantitatively estimate various performance attributes of target-detection WSNs. While ANDES derives models for the average detection delay and the probability of detection for duty cycled WSNs, it includes neither energy-based joint optimization nor modeling for off-the-shelf communication protocols. In addition, the designers of ANDES make the strong assumption of a uniformly distributed deployment, whereas the optimization toolkit proposed in the following holds for a wider variety of node placements. The strategies and techniques for node placement are covered in [13,14], where the authors propose a taxonomy of deployments (e.g., random vs. deterministic, static vs. dynamic role attribution) and review the literature. For instance, in [13] the authors provide some insights about random/deterministic node placement strategies to guarantee a minimum network coverage. However, these approaches do not provide a performance description in terms of probability of missed target detection, when sensing duty cycles are considered. While we here undertake a pragmatic approach, considering the impact of a few arbitrary node placements on the predicted performance, in [15] the optimization of node placement, with respect to area coverage or network connectivity and taking into account the dynamic allocation of the sink role among the nodes, is considered.

The design of target detection applications represents also a very active area of research. Following an implementation-oriented approach, in [5] the authors present the design and implementation of a monitoring system, referred to as VigilNet, based on a WSN with the Crossbow MicaZ platform [16]. They derive an energy-efficient adaptive surveillance strategy and validate it through experimental tests. In [17], under the assumptions that the road network is known and the target movement is confined into roads, the authors describe an algorithm, referred to as VIrtual Scanning Algorithm (VISA), which guarantees that the incoming target will be detected before reaching a given protection point. In [18], the benefits brought by the use of mobile nodes for target detection are evaluated. In [19], the authors introduce a duty cycling strategy, using magnetic sensors, for power-efficient and reliable target detection.

While the related works deal with specific aspects of wireless sensor networking for target detection, our approach considers a WSN-based system for mobile target detection as a whole, clearly highlighting relevant cross-layer interactions.

3. Sensing layer: Probability of target missed detection

In this section, we concentrate on the effect of *sensing* duty cycling on the probability of target missed detection. Assuming that a network operator deploys the available nodes in known positions (e.g., in the proximity of crossings, building perimeter, etc.), the problem reduces to assessing the probability of target missed detection as a function of the nodes' configuration (e.g., the sensing duty cycle), their number, and their relative positions. In this context, we derive upper and lower bounds for the probability of missing an incoming target. These bounds allow a network designer to evaluate the effectiveness of a specific node deployment for monitoring a critical area of interest. Once taken into account in a complete cross-layer framework, the probability of missed detection will contribute to an accurate cross-layer system perspective overview.

3.1. Preliminary system assumptions

The wireless sensor nodes considered in this paper are equipped with a seismic sensor, whose sensing range is denoted as r_s (dimension: [m]). To reduce the energy consumption of the system, the sensing part can be periodically switched off, according to a normalized duty cycle $\beta_{sens} \in [0,1]$ over a period t_{sens} (dimension: [s]). More precisely, nodes sense the surrounding environment for an interval of length $\beta_{sens}t_{sens}$ and sleep for an interval of duration $(1-\beta_{sens})t_{sens}$. This sensing/sleeping pattern repeats cyclically. We assume that all sensors have the same values of r_s, β_{sens}, and t_{sens}. We assume that the sensing duty cycles at the nodes are not synchronized. This assumption holds for target detection purposes, as a node only needs to have its sensing module on. At the opposite, in the communication phase, i.e., when a node is transmitting a packet to a neighboring node, both transmitting and receiving nodes must be on at the same time, thus requiring a sort of synchronization between them.

To make the derivation of the probability of target detection P_d (or, equivalently, of the probability of target missed detection P_{md}) feasible, we assume the monitored area to be a square with sides of length d_s (dimension: [m]). In this area, N sensors are placed in *known* positions, under the constraint that their sensing ranges do not overlap. This constraint is reasonable for surveillance applications where WSNs are deployed by network operators: in fact, overlapping sensed areas would reduce the system detection capabilities. In addition, each node must have at least another node at a distance shorter than the transmission range, denoted as r_T (dimension: [m]), thus guaranteeing that it can transmit a packet to at least a neighboring node.

We assume that the potential targets cross the monitored area following linear trajectories. An illustrative example is shown in Fig. 1. Each trajectory is characterized by (i) an entrance angle, with respect to a reference axis given by the entrance side, uniformly distributed in $[0,\pi]$; and (ii) a constant target speed v (dimension: [m/s]). Since there is no information about the entrance point, we also assume that the target entrance point into the monitored area is uniformly distributed over the perimeter of the monitored surface.

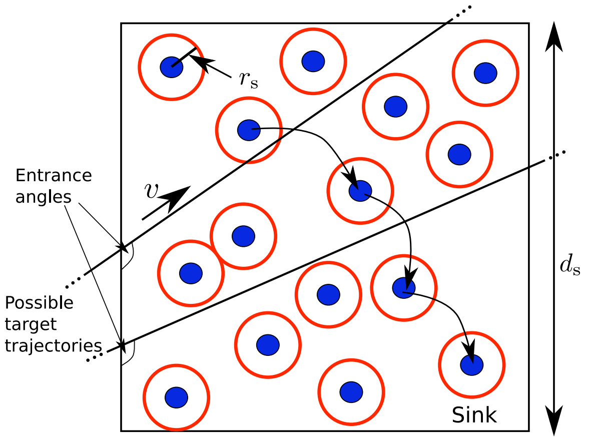


Fig. 1. Illustrative example of the WSN-based detection system of interest, with two possible target trajectories.

Table 1
Constant and variable parameters considered in the simulations. The default values (used in the remainder of this paper) are indicated between parentheses.

Constant parameters		
Side of monitored area	d_s	1000 m
Speed of the target	v	15 m/s
Transmission range	r_T	250 m
Sensing period	t_{sens}	15 s
Variable parameters (with default value ranges)		
Number of nodes in the network	N	5–25 (20)
Sensing range of each node	r_s	20, 50 (20) m
Sensing duty cycle	β_{sens}	0.1–1

The main model parameters introduced above are set to the values listed in Table 1.

3.2. Probability of target missed detection

In the *stochastic* node deployment scenario considered in [6], where the nodes are uniformly distributed over the monitored area and their positions are not a priori known, all sensing nodes are independent and the probability of missed detection simply reduces to the evaluation of the probability that a single sensor misses the target. In the *deterministic* node deployment scenario considered here, the nodes are not independent and identically distributed over the monitored area but, rather, have specific (a priori known) positions. In order to evaluate the probability of missed detection, one needs to compute the probability that none of the sensors detects the target. This computation is analytically unfeasible, as it would require the evaluation of all possible trajectories (namely, their intersections with the union of the sensed areas). However, the derivation of upper and lower bounds is analytically tractable. In the following, we first derive these bounds in the absence of duty cycling and, then, extend it by incorporating sensing duty cycling.

3.2.1. Absence of sensing duty cycling

Given N sensors placed in known positions, the probability of target detection, defined as the probability that a linear trajectory across the monitored area crosses at least one node sensed area, can be expressed as [20]

$$P_d = 1 - P_{md} = P(\ell \cap \mathscr{A}_1 \cup \mathscr{A}_2 \cup \cdots \cup \mathscr{A}_N), \tag{1}$$

where ℓ is a generic line crossing the monitored surface, and $\mathscr{A}_1, \ldots, \mathscr{A}_N$ are the sensed areas. The expression at the right-hand side of (1) can be rewritten, using the Feller's inclusion–exclusion principle [21], as the sum of joint probabilities of a line intersecting specific set arrangements:

$$P_d = \sum_{i=1}^{N} P(\ell \cap \mathscr{A}_i \neq \emptyset) - \sum_{i,j:i<j}^{N} P(\ell \cap \mathscr{A}_i \cap \mathscr{A}_j \neq \emptyset) + \cdots + (-1)^N P(\ell \cap \mathscr{A}_1 \cap \mathscr{A}_2 \cdots \mathscr{A}_N), \tag{2}$$

where $P(\ell \cap \mathscr{A}_i \neq \emptyset)$ denotes the probability that the linear trajectory crosses the ith sensed area.

The expression at the right-hand side of (2) is difficult to evaluate, since it requires information on the probability that a generic line crosses all sensors' subsets. For instance, considering the first term at the right-hand side of (2), the subset is formed by one sensor, whereas, considering the last term, the subset is formed by all N nodes in the network. Therefore, the following upper and lower bounds can be simply obtained by considering only a few terms among those in the expression at the right-hand side of (2):

$$\sum_{i=1}^{N} \underbrace{P(\ell \cap \mathscr{A}_i \neq \emptyset)}_{\triangleq m_1(i)} - \sum_{i,j:i<j}^{N} \underbrace{P(\ell \cap \mathscr{A}_i \cap \mathscr{A}_j \neq \emptyset)}_{\triangleq m_2(i,j)} < P_d < \sum_{i=1}^{N} \underbrace{P(\ell \cap \mathscr{A}_i \neq \emptyset)}_{=m_1(i)}, \tag{3}$$

where $m_1(i)$ is the probability that a generic line crosses the area sensed by node i. Obviously, the larger is the used number of additive terms from the *exact* expression at the right-hand side of (2), the more accurate are the upper and lower bounds in (3).[2] Being the area sensed by a node circular, on the basis of the framework in [20] it can be shown that

$$m_1(i) = \frac{2\pi r_s}{4d_s} \quad \forall i.$$

The term $m_2(i,j)$ in the lower bound in (3), instead, represents the probability that the target trajectory crosses both the areas sensed by nodes i and j. In this case, at least one of the nodes i and j must be active when the target is crossing its sensed area, in order to detect the target. Its computation, unlike that of m_1, requires information about the (relative) node positions. In Appendix A, it is shown that $m_2(i,j)$ can be given the expression (A.1).

[2] Note that each additive term in (2) is smaller, in absolute value, of the previous one. In fact, consecutive terms correspond to the sum of the probabilities of the intersections between a larger and larger number of sets.

3.2.2. Presence of sensing duty cycles

We consider the sensing duty cycle illustrated in Fig. 2: the sensing unit of each node is active for $100\beta_{\text{sens}}\%$ of a period of duration t_{sens}, and off for $100\,(1-\beta_{\text{sens}})\%$ of the period. The absence of sensing duty cycling, considered in the previous subsection, corresponds to the case where the sensing unit of each node is always switched on, i.e., to the case with $\beta_{\text{sens}} = 1$. In the following, we introduce the following events:

$$\mathcal{E}_{\text{det}}^{(i)} \triangleq \{\text{The target is detected by node } i\},$$

$$\mathcal{E}_{\text{det}}^{(i)} \triangleq \{\text{The target is detected by at least one of the nodes } i \text{ and } j\},$$

$$\mathcal{E}_{\text{SoT}_i} \triangleq \{\text{The target's trajectory crosses the area sensed by node } i\},$$

$$\mathcal{E}_{\text{SoT}_{i,j}} \triangleq \{\text{The target's trajectory crosses both areas sensed by nodes } i \text{ and } j\}.$$

The bounds (3) can then be directly extended, to encompass the presence of sensing duty cycles, as follows:

$$\sum_{i=1}^{N} m_1(i) P\left\{\mathcal{E}_{\text{det}}^{(i)} | \mathcal{E}_{\text{SoT}_i}\right\} - \sum_{i,j:i<j}^{N} m_2(i,j) P\left\{\mathcal{E}_{\text{det}}^{(i,j)} | \mathcal{E}_{\text{SoT}_{i,j}}\right\} < P_{\text{d}} < \sum_{i=1}^{N} m_1(i) P\left\{\mathcal{E}_{\text{det}}^{(i)} | \mathcal{E}_{\text{SoT}_i}\right\}, \qquad (4)$$

where $P\left\{\mathcal{E}_{\text{det}}^{(i)} | \mathcal{E}_{\text{SoT}_i}\right\}$ is the probability that the target is detected by node i, *given that* the target crosses its sensed area; and $P\left\{\mathcal{E}_{\text{det}}^{(i,j)} | \mathcal{E}_{\text{SoT}_{i,j}}\right\}$ is the probability that the target is detected by at least one of the nodes i and j, *given that* the target crosses the areas sensed by both nodes. In Appendix B, it is shown that $P\left\{\mathcal{E}_{\text{det}}^{(i)} | \mathcal{E}_{\text{SoT}_i}\right\} = P\{\mathcal{E}_{\text{det}} | \mathcal{E}_{\text{SoT}}\}, \forall i$, and an expression for this probability is derived. In Appendix C, an expression for $P\left\{\mathcal{E}_{\text{det}}^{(i,j)} | \mathcal{E}_{\text{SoT}_{i,j}}\right\}$, $\forall i,j : i<j$ is derived. From (4), it is straightforward to derive the following bounds for the probability of target *missed detection*:

$$1 - \frac{2\pi r_{\text{s}} N}{4 d_{\text{s}}} P\{\mathcal{E}_{\text{det}} | \mathcal{E}_{\text{SoT}}\} < P_{\text{md}} < 1 - \frac{2\pi r_{\text{s}} N}{4 d_{\text{s}}} P\{\mathcal{E}_{\text{det}} | \mathcal{E}_{\text{SoT}}\} + \sum_{i,j:i<j}^{N} m_2(i,j) P\left\{\mathcal{E}_{\text{det}}^{(i,j)} | \mathcal{E}_{\text{SoT}_{i,j}}\right\}. \qquad (5)$$

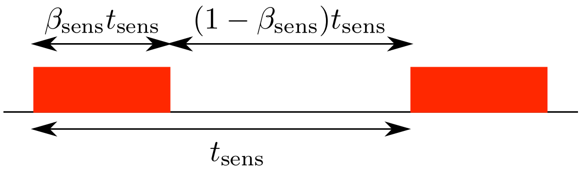

$\beta_{\text{sens}} t_{\text{sens}}$
$(1-\beta_{\text{sens}}) t_{\text{sens}}$
t_{sens}

Fig. 2. Logical scheme of the sensing duty cycle.

3.3. Performance evaluation

We now analyze the accuracy of the proposed analytical framework for evaluation of the probability of target missed detection by comparing the predicted performance with realistic simulation results. According to the simulator implementation,[3] we consider node deployments over a square area and use, for the system parameters, the values introduced in Section 3.1. In each simulation run, a target enters the monitored area from one of the sides (randomly selected) with an entrance angle uniformly distributed between 0 and π (with respect to the entrance side). The target moves with a constant speed v. If the target crosses a sensed area while the sensing device is on, the target is detected. Otherwise, if the target exits the monitored area without being detected by any node, the target is declared lost. To evaluate the probability of target missed detection, we average the simulation results over 1000 different topologies and, for each topology, we consider 1000 different target trajectories (each trajectory is associated with randomly generated entrance point and angle).

In Fig. 3, the probability of missed detection is shown as a function of β_{sens}. Considering the curves relative to $N = 5$, the upper and lower bounds are close to each other. In fact, since there are only a few nodes in the network, it is likely that a target crosses only one or, at most, two sensed areas. In this case, the performance is well approximated by using the terms $\{m_1(i)\}$ and $\{m_2(i,j)\}$, which can be computed as described in Section 3.2. When β_{sens} is small, the simulated values of P_{md} are slightly below the lower bound. In this case, the lower bound—based only on the use of the term $\{m_1(i)\}$—is not sufficiently accurate to well approximate the true value of P_{md}. In order to improve the accuracy of the lower bound, the computation of the contribution of the higher-order terms in expression (2) would be required. However, from a network designer perspective, it may be sufficient that the upper bound is accurate. More precisely, one can minimize the upper bound for P_{md}, thus guaranteeing that, for each considered topology, the effective probability of missed detection will be lower than that predicted by the upper bound.

Considering the case with $N = 10$ nodes, instead, the simulated performance lies inside the bounds for all values of β_{sens}. In this case, however, the lower bound is quite coarse. However, the upper bound remains quite close to the simulation performance. In fact, the presence of the terms $\{m_2(i,j)\}$ allows to better approximate P_{md}. When $N = 25$ nodes are considered in Fig. 3, it can be observed that the simulation performance lies inside the bounds for $\beta_{\text{sens}} \geqslant 0.2$. While the upper bound tends to become loose for high values of β_{sens}, the lower bound gets tight. In fact, it is more likely that there are three or more sensed areas crossing the target trajectory, so that the computation of only the terms $\{m_1(i)\}$ is not sufficient to correctly estimate P_{md}. Thus, the terms $\{m_2(i,j)\}$ play a key role and the LB becomes more accurate than the UB.

[3] The simulator, available upon request, has been developed by the authors in C.

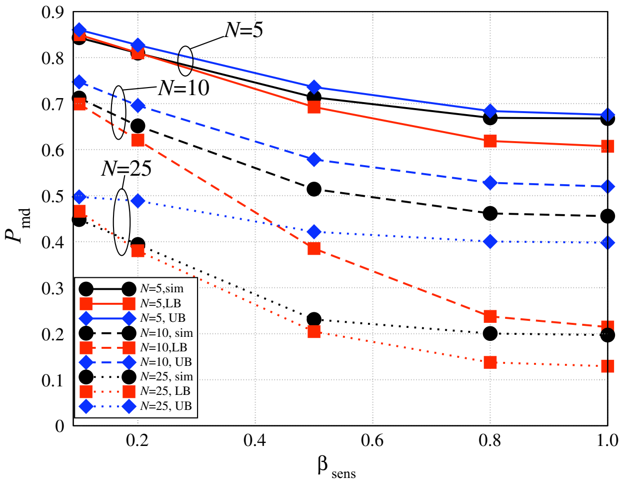


Fig. 3. P_{md} as a function of β_{sens}, considering various values for the number N of deployed sensor nodes. Simulation (sim) and analytical lower/upper bounds (LB/UB) are shown.

Note that the floor on P_{md}, asymptotically reached when $\beta_{\text{sens}} = 1$, can be analytically evaluated with the following discretized approach. Recall preliminary that the entrance point can lie anywhere on the perimeter of the monitored area and for each entrance point, the entrance angle, with respect to the entrance side, can vary in $[0, \pi]$. The (infinite) set of all possible trajectories can be discretized considering linear steps, of width Δ_x, over the perimeter and, for each step, angular steps of width Δ_θ. This leads to a set of $N_{\text{traj}} \triangleq N_x \times N_\theta$ trajectories, where $N_x \triangleq 4d_s/\Delta_x$ and $N_\theta \triangleq \pi/\Delta_\theta$. Given a specific topology realization, for each possible discretized trajectory we check if it crosses at least a sensed area. Denoting by N_{cross} the number of trajectories which cross at least a sensed area, the asymptotic floor (i.e., the value of P_{md} in correspondence to $\beta_{\text{sens}} = 1$) can be simply estimated as $N_{\text{cross}}/N_{\text{traj}}$. In Table 2, the estimated floor, obtained considering $\Delta_x = 20$ m and $\Delta_\theta = \pi/30$, is compared with the simulation values results shown in Fig. 3. By considering smaller values of Δ_x and Δ_θ, the accuracy of the estimated value of P_{md} can be increased.

Due to the dependence of P_{md} on the node positions (i.e., the specific topology realization), it is of interest to analyze the impact of the node spatial distribution. To do this, we consider the average distance, denoted as $\bar{d}_{\text{pair}}$, between all possible pairs of nodes. For each generated topology and associated value of $\bar{d}_{\text{pair}}$, we compute the upper and lower bounds and check if the simulation-based (averaged out over 1000 runs) value of P_{md} lies between the bounds. We consider 1000 topologies and count, for each value of $\bar{d}_{\text{pair}}/d_s$ (quantized in intervals of length 0.03), the number of cases where the obtained performance lies/does not lie between the bounds. The results, in terms of percentage of times that simulation results lie/do not lie within the bounds are shown, as functions of the ratio $\bar{d}_{\text{pair}}/d_s$, in Fig. 4a and b, respectively. As one can see, the simulated probability of missed detection lies outside the bounds only in a few cases, confirming the validity of the proposed analytical framework.

Table 2
Asymptotic ($\beta_{\text{sens}} = 1$) values of P_{md} (i) estimated using a discretized trajectory approach and (ii) evaluated by simulations.

N	Estimated P_{md}	Simulated P_{md}
5	0.7251	0.6676
10	0.5561	0.45544
25	0.3199	0.1971

On the basis of other results (not presented here for conciseness), the following conclusions can be carried out.

- For small values of N, the number of topologies, whose simulation-based value of P_{md} is outside the bounds, is larger. In particular, by considering the node spatial density $\rho_s \triangleq N/d_s^2$ (dimension: $[\text{m}^{-2}]$), it can be concluded that, for $\rho_s > 5/1000^2 = 5 \times 10^{-6}$ nodes/m^2, the framework allows to accurately predict the probability of target missed detection in the considered scenarios with $r_s = 50$ m.
- For small values of β_{sens}, it is more likely that P_{md} lies outside the bounds.
- Considering only the topologies whose performance lies below the upper bound, the approximation works better, allowing a good performance prediction.

4. Sensing layer: Delay before first sensor detection

In this section, we derive an analytical model for the evaluation of the average delay before the first sensor detection, denoted as $\overline{D}_{\text{det}}$, defined as the average time interval between the instant at which a target enters the monitored area and the time at which it is first detected by a sensor.

4.1. Absence of sensing duty cycling

In the absence of sensing duty cycling, i.e., with sensing units always on, and *under the assumption that any trajectory crosses at least a sensed area*, the average delay before the first sensor detection can be expressed as follows:

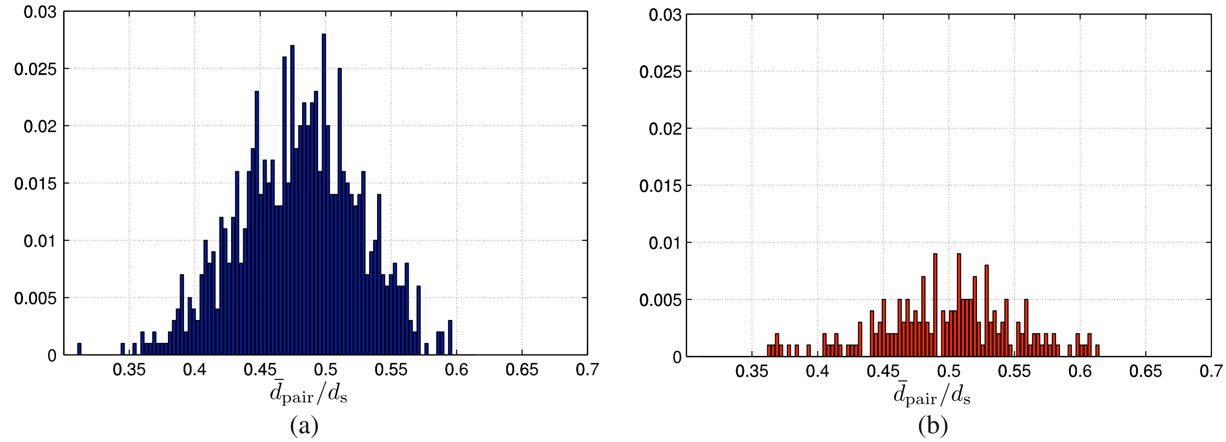


Fig. 4. Percentage of times, for possible values of the ratio $\bar{d}_{\mathrm{pair}}/d_{\mathrm{s}}$, that the simulated probability of missed detection lies (a) inside or (b) outside the bounds. The number of nodes is $N = 10$.

$$\overline{D}_{\mathrm{det}} = \int_{\mathscr{D}_\Theta}\int_{\mathscr{D}_X} D(x,\theta) f_{X,\Theta}(x,\theta)\mathrm{d}x\,\mathrm{d}\theta, \qquad (6)$$

where X, whose specific realization is x, denotes the entrance point (over the perimeter of the square area) of the target and Θ, whose specific realization is θ, denotes the entrance angle (with respect to the side from which the target is entering the area) of the target. Owing to the randomness of the trajectory, X and Θ are independent and, therefore,

$$f_{X,\Theta}(x,\theta) = f_X(x) f_\Theta(\theta),$$

where $X \sim \mathrm{Unif}[0, 4d_\mathrm{s}]$ ($\mathscr{D}_x = [0, 4d_\mathrm{s}]$, i.e., perimeter of the monitored area); $\Theta \sim \mathrm{Unif}[0,\pi](\mathscr{D}_\theta = [0,\pi])$; and $D(x,\theta)$ is the exact delay (before hitting the first sensed area[4]) associated with the specific target trajectory identified by x and θ.

While expression (6) for the average delay holds provided that *any* trajectory can be detected, in reality it may happen that there are undetectable trajectories (see, for example, the lower trajectory in Fig. 1). In this case, the direct application of expression (6) is not possible, as the delay of an undetectable trajectory would be infinite. Therefore, the average delay before the first detection act is a concept which does not apply to an undetectable trajectory. In other words, only detectable trajectories are meaningful for the evaluation of the delay before the first detection act. More precisely, the average delay can be evaluated by taking into account that for each entrance point x there is an angular interval, denoted as $\mathscr{D}_\theta(x) \subset [0,\pi]$, given by all entrance angles which correspond to detectable trajectories. The delay before first detection can then be written as

$$\overline{D}_{\mathrm{det}} = \int_0^{4d_\mathrm{s}} \frac{1}{4d_\mathrm{s}} \int_{\mathscr{D}_\theta(x)} \frac{1}{|\mathscr{D}_\theta(x)|} D(x,\theta)\mathrm{d}\theta\,\mathrm{d}x, \qquad (7)$$

where $|\mathscr{D}_\theta(x)| \leqslant \pi$ is the "length" of the angular interval $\mathscr{D}_\theta(x)$.

According to the notation already introduced in Section 3.3, the integral expression (7) for the average delay can be numerically evaluated by considering discretized integration steps, denoted as Δ_x and Δ_θ, for the entrance position (along the perimeter) and angle (with respect to the entrance side). Therefore, the average delay before first detection can be approximated as follows:

$$\overline{D}_{\mathrm{det}} \simeq \frac{1}{N_x}\sum_{i=0}^{N_x} \frac{1}{N_\theta(i\Delta_x)} \sum_{j=0}^{N_\theta(i\Delta_x)} D(i\Delta_x, j\Delta_\theta), \qquad (8)$$

where $N_x = 4d_\mathrm{s}/\Delta_x$ is the *fixed* number of discretized linear steps over the perimeter and $N_\theta(i\Delta_x)$ is the number of admissible (i.e., corresponding to detectable trajectories) discretized angular steps, of *fixed* width Δ_θ, in correspondence to the ith discretized entrance point along the perimeter.[5] By considering sufficiently small values of Δ_x and Δ_θ, the accuracy of the numerical evaluation of the integral (7) can be increased as desired.

4.2. Presence of sensing duty cycling

The final expression (7) for the average delay before the first detection holds when the sensing interface is always on, i.e., for $\beta_{\mathrm{sens}} = 1$. In this case, if a trajectory crosses at least one sensed area, then the target is detected for sure. At the opposite, when the the sensing interfaces of the nodes are duty cycled, even if the trajectory crosses at least a sensed area, it may happen that none of the sensors detects the target. In this case, the concept of delay is not defined, as the delay is a meaningful performance metric *given that* the target is detected—also from a practical perspective, it would be impossible to compute it. Therefore, the analysis in Section 4.1 has to be extended under the assumption of target detection.

[4] Note that the delay $D(x,\theta)$ is simply given by the ratio between the distance from the entrance point and the perimeter of the first hit sensed area and the speed v of the target.

[5] We remark that while N_x is fixed, regardless of the entrance point along the perimeter, $N_\theta(i\Delta_x)$ depends on the specific entrance point and the current WSN topology realization.

Expression (7) for the delay can be extended by observing that (i) along an admissible trajectory, identified by (x,θ), there might be more than one sensor and, therefore, (ii) the delay depends on which sensor actually detects the target. In general, one can write:

$$\overline{D}_{\mathrm{det}} = \int_0^{4d_s} \frac{1}{4d_s} \int_{\mathscr{D}_\theta(x)} \frac{1}{|\mathscr{D}_\theta(x)|} \overline{D}(x,\theta)\,\mathrm{d}\theta\,\mathrm{d}x, \tag{9}$$

where $\overline{D}(x,\theta)$ is the average detection delay along the trajectory identified by (x,θ). In Appendix D, it is shown that

$$\overline{D}(x,\theta) = \sum_{h=1}^{n(x,\theta)} D_h(x,\theta) \times \frac{P\left\{\mathscr{E}^{(h)}_{\mathrm{det}}|x,\theta\right\}\prod_{k=1}^{h-1}P\left\{\overline{\mathscr{E}}^{(k)}_{\mathrm{det}}|x,\theta\right\}}{\sum_{z=1}^{n(x,\theta)}P\left\{\mathscr{E}^{(z)}_{\mathrm{det}}|x,\theta\right\}\prod_{v=1}^{z-1}P\left\{\overline{\mathscr{E}}^{(v)}_{\mathrm{det}}|x,\theta\right\}}, \tag{10}$$

where $D_h(x,\theta)$ is the delay when the hth sensor is the first to detect the target and $P\left\{\mathscr{E}^{(h)}_{\mathrm{det}}|x,\theta\right\}$ is the probability that the hth sensor along the trajectory (x,θ) detects the target.

As in the absence of sensing duty cycling, the average delay before detection can be numerically approximated as follows:

$$\overline{D}_{\mathrm{det}} \simeq \frac{1}{N_x}\sum_{i=0}^{N_x}\frac{1}{N_\theta(i\Delta_x)}\sum_{j=0}^{N_\theta(i\Delta_x)}\sum_{h=1}^{n(i\Delta_x,j\Delta_\theta)} D_h(i\Delta_x,j\Delta_\theta) \cdot \frac{P\left\{\mathscr{E}^{(h)}_{\mathrm{det}}|i\Delta_x,j\Delta_\theta\right\}\prod_{k=1}^{h-1}P\left\{\overline{\mathscr{E}}^{(k)}_{\mathrm{det}}|i\Delta_x,j\Delta_\theta\right\}}{\sum_{z=1}^{n(i\Delta_x,j\Delta_\theta)}P\left\{\mathscr{E}^{(z)}_{\mathrm{det}}|i\Delta_x,j\Delta_\theta\right\}\prod_{v=1}^{z-1}P\left\{\overline{\mathscr{E}}^{(v)}_{\mathrm{det}}|i\Delta_x,j\Delta_\theta\right\}}, \tag{11}$$

where N_x and $N_\theta(i\Delta_x)$ are defined as at the end of Section 4.1.

4.3. Performance evaluation

We now investigate the average delay before first detection. More precisely, together with the analytical framework developed in the previous subsections, we also evaluate the delay by simulations, using the same simulator presented in Section 3.3. In particular, the average performance delay is evaluated, through simulations, as follows. Among all randomly generated trajectories, we consider only those which lead to target detection. Then, we simply consider the arithmetic average of the detection delays observed in these cases. The sensing range is set to $r_s = 50$ m, whereas the other system parameters are set as indicated in Table 1.

In Fig. 5, $\overline{D}_{\mathrm{det}}$ is shown as a function of β_{sens}, considering various values of the number N of sensors. As one can see, regardless of the value of N the average delay is a decreasing function of β_{sens}. This is expected, as increasing β_{sens} increases the probability of detection by each sensor and, therefore, it becomes more likely that "early" sensors along a detectable trajectory detect the target. It can be observed that while there is a non-negligible discrepancy between simulations and analysis for the case with $N = 5$, the agreement becomes good for increasing values of N. In all cases, however, the trend predicted by the analysis is confirmed by the simulation results. Moreover, it can be observed that the delay basically remains constant for $\beta_{\mathrm{sens}} \geqslant 0.8$.

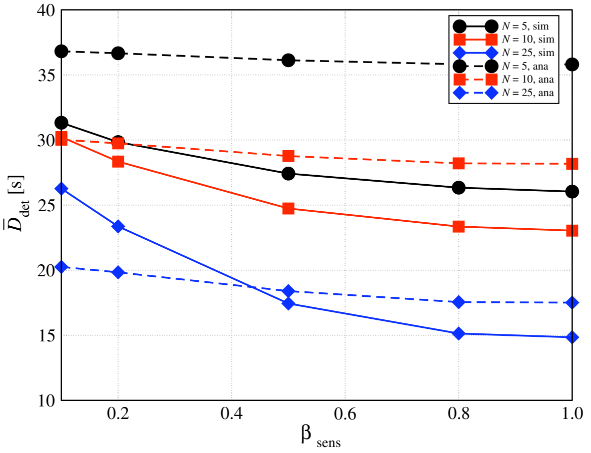


Fig. 5. Average delay before first target detection as a function of β_{sens}. Simulation (sim) and analytical (ana) results are presented for various values of N.

As discussed at the beginning of Section 4.2, the delay before first detection is a concept applicable only to the cases where the target can be actually detected. In particular, in the simulations only the trajectories, where the target *is* detected, are considered. In this sense, it can be a misleading performance indicator, as it is not representative of the probability of detecting a target—in other words, detecting 1% of the targets in a very short time might not be useful in a surveillance system. Therefore, the delay before detection becomes a more meaningful performance indicator when correlated with the corresponding probability of detection. In Fig. 6, we show the average delay

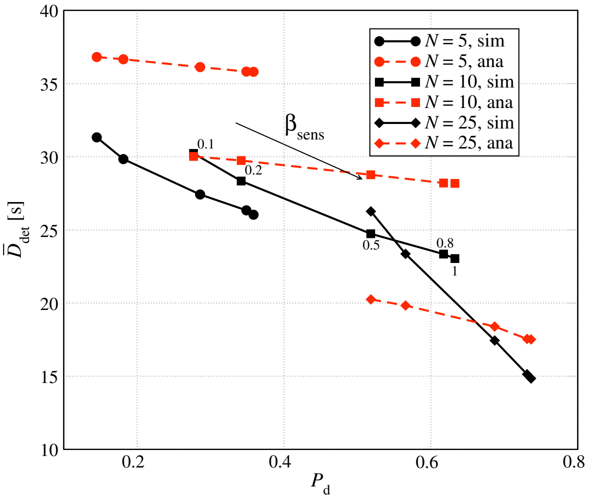


Fig. 6. Average delay before first detection as a function of the probability of detection (curves parametrized in terms of β_{sens}). Simulation (sim) and analytical (ana) results are presented for various values of N.

(predicted by the analysis and by the simulator) as a function of the probability of detection, considering various values of N. In the analytical case, the probability of detection is estimated as the arithmetic average of the upper and lower bounds in (4). For each value of N, both simulation (solid lines) and analytical (dashed lines) results have been obtained by parameterizing in terms of β_{sens} the results in Figs. 3 and 5. It can be observed that for increasing values of β_{sens} the delay reduces and the probability of detection increases. In other words, the two performance indicators improve simultaneously for increasing values of β_{sens}. Therefore, this implies that in the proposed WSN-based surveillance systems the two indicators cannot be optimized independently of each other.

5. Communication layer: Latency after detection

Whenever an event of interest is detected, sensors communicate an alert to the sink using the radio channel. This communication happens in a single or multihop fashion depending on the relative locations of the sensing node and the sink. We call *latency after detection* the time interval between the instant of first detection of an event by a sensor and the instant of its notification to the sink. The latency is an important indicator of the performance of a WSN-based detection system, as it is representative of the reactivity with which an operator would become aware of an alert. In real-time applications, the WSN should guarantee that latency is shorter than a maximum tolerable value.

There are a number of parameters that influence the delay to transmit a message from node to node, such as radio duty cycling, propagation time, or processing time. In the following, we make a few simplifying assumptions to derive a simple, yet accurate, analytical model of the latency. In particular, we consider that radio duty cycling and channel access control represent the most significant sources of delay, with respect to which processing and propagation times can be neglected. As duty cycling was considered at the sensing layer, it can also be considered at the communication layer. In fact, it is a widely used communication mechanism in WSNs. More precisely, it refers to the activation and deactivation of the radio chip interface, for energy-saving purposes, on the basis of sleeping and waking-up schedules. Communication duty cycling is typically in charge of the MAC protocol. When active, a node is able to transmit or receive data; when sleeping, the node completely turns off its radio to save energy.

Many schemes have been proposed in the literature to mitigate the latency caused by duty cycled MAC protocols. One can distinguish between the asynchronous approaches, where nodes set the sleep/wake-up schedules in a fully independent fashion, and the synchronous ones, where there is some kind of alignment. In [6], a latency model based on the asynchronous X-MAC protocol [1] is proposed. In the current paper, we propose another latency model based on the novel (synchronous) Cas-MAC protocol. Although the latter protocol shares the principle of operations of the D-MAC protocol [2] (i.e., cascading the duty cycles), it goes many steps forward in terms of design, implementation, and performance modeling. The Cas-MAC protocol design is tailored to meet the dominant traffic pattern encountered in a target detection system, i.e., traffic over a data gathering tree, in order to minimize even further the delay at the price of little synchronization overhead.

We first recall the analytical latency model derived in [6] for the X-MAC protocol, and then introduce the Cas-MAC protocol and its associated analytical latency model. Both analytical models will be validated with experimental results.

5.1. An asynchronous MAC protocol: X-MAC

The X-MAC protocol uses Low-Power Listening (LPL), or preamble sampling, to enable low-power communications between a sender and a receiver which do not synchronize their wake-up and sleep schedules. The X-MAC protocol uses a *strobed preambling* approach in which a sender with data quickly alternates between the transmission of the destination address and a short waiting time, so that the receiver can potentially acknowledge that it is ready to receive data. The preambling phase lasts for at most the duration of the sleeping interval.[6]

This approach allows to reduce energy consumption and the per-hop latency with respect to protocols using long-preambles such as, for example, B-MAC [23]. In [22], the authors show that the X-MAC protocol performs at least as well as some of well-known state-of-art MAC protocols, including the Scheduled Channel Polling (SCP-MAC) protocol [24], standard TDMA [25], or B-MAC, in terms of latency and throughput.

The average transmission latency per hop can be expressed as

$$D_{\text{X-MAC}_{1\,\text{hop}}} = \frac{(1-\beta_{\text{comm}})^2 t_{\text{comm}}}{2} + S_{\text{p}} + S_{\text{al}} + S_{\text{d}}, \tag{12}$$

where β_{comm} is the (normalized) communication duty cycle over the period t_{comm}, and S_{p}, S_{al}, S_{d} are the durations (constant values, dimension: [s]) of the strobed preamble, the acknowledgment of the preamble, and the alert packet, respectively. Further details about the derivation of expression (12), obtained averaging over the worst and best transmission cases, can be found in [6].

Considering a multi-hop path, the average global latency can be expressed as follows:

$$D_{\text{X-MAC}} = D_{\text{X-MAC}_{1\,\text{hop}}} N_{\text{hop}}, \tag{13}$$

where N_{hop} denotes the average number of hops that the alert message traverses to reach the sink and depends on the network topology. Various relevant examples of WSN topologies will be considered in Section 7.

5.2. A synchronous MAC protocol: Cas-MAC

As shown in [2], the most significant traffic pattern in a WSN for target detection is data gathering from the

[6] For experimental validation, we have used the implementation of X-MAC protocol provided by UPMA/MLA [22]. In the latter, the sender preambles the data packet itself along with the destination address so that the receiver acknowledges the reception of the data packet.

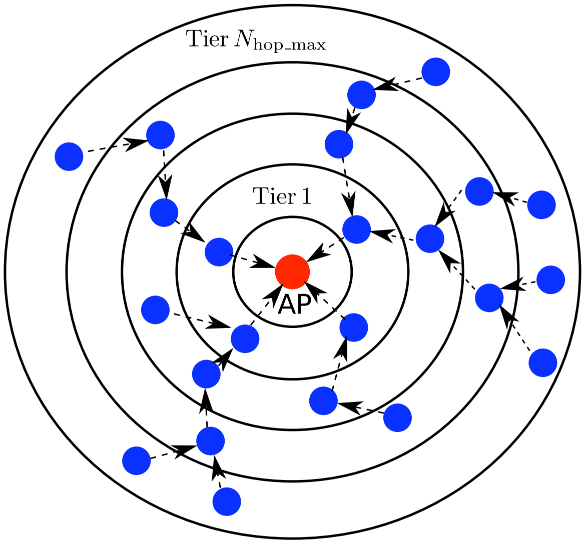


Fig. 7. Illustrative example of a WSN scenario with tree topology.

sources to the sink. In Fig. 7, an illustrative scenario of a WSN with tree topology is shown. In this case, all sensors transmit to a common sink in a multi-hop fashion along unidirectional paths that are likely to remain stable for a sufficiently long time. In [2], the authors present the D-MAC protocol, which implements a staggered wake-up schedule on multi-hop paths in order to wake up nodes sequentially like a chain reaction. This approach has the key advantage that the one-hop latency roughly reduces to the offset introduced between the duty cycles, regardless of the length t_{comm} of a duty cycle. In [2], an ns-2 simulator for analyzing the performance of the D-MAC protocol is proposed.

The Cas-MAC protocol is inspired by the D-MAC protocol. In [2], in the D-MAC protocol implementation for the ns-2 simulator, it is assumed that nodes are synchronized. However, the provision of a synchronization capability is a challenging task in duty cycled WSNs, as its implementation can dramatically degrade the performance of the protocol. With the Cas-MAC protocol, a step further, with respect to the the general design principles of the D-MAC protocol, is taken. In fact, the Cas-MAC protocol makes critical choices on the actual implementation of the synchronization and transmission routines. The main design aspects, illustrated in Fig. 8, can be summarized as follows.

- *Skewing of duty cycles*: we assume that all nodes have the a priori knowledge of how deep they are in the WSN data gathering tree. At regular intervals, an explicit synchronization phase occurs where the sink node broadcasts to all nodes its wake-up and sleep schedule. Knowing its depth in the tree, denoted as[7] d, each node offsets its own duty cycle, with respect to the schedule of the sink, by $d \cdot \xi$, where the skew ξ is defined as the one-hop temporal offset introduced between the schedules of two subsequent nodes on the path to the sink. The skewing of duty cycles is repeated periodically to adjust possible drifts in the internal clocks of the nodes. The synchronization phase is carried out by letting the sink node announce repeatedly its next wake-up time for an interval at least as long as the duty cycle length. In this way, one guarantees that all other nodes will wake up and poll the channel for a sufficiently long time to receive at least one packet from the sink. If needed, the nodes will skew their duty cycle according to their depth in the tree.
- *Sending/relaying a packet*: in every duty cycle at each node, an alarm fires just before the next node along the multi-hop path to the sink wakes up. At that time, the former node checks whether a buffered packet has to be sent to the next node. If so, the node first sends a wake-up tone and, then, sends the payload packet in unicast to the next node.
- *Contention handling*: we use a two-step contention phase as in the SCP-MAC protocol [26]. While it does not fully prevent the hidden node problem, it reduces significantly the risk of collision that may occur at the junction of tree branches.

The average transmission delay for the first hop of the transmission chain can be expressed as

$$D_{\text{Cas-MAC}_{\text{1st hop}}} = \frac{t_{\mathrm{comm}}}{2} + CW + WT_{\mathrm{length}} + S_{\mathrm{d}}, \tag{14}$$

where $t_{\mathrm{comm}}/2$ designates the average time the node with data has to wait before the internal alarm fires; CW (dimension: [s]) is the average overall duration of the contention windows; and WT_{length} (dimension: [s]) is the duration of the wake-up tone.

In the remaining $N_{\mathrm{hop}} - 1$ hops to the sink, the sending routine (i.e., wake-up tone, two-step contention) of the packet starts once the alarm fires. Therefore, from the second hop on the one-hop delay approximates the value of the duty cycle offset. The multi-hop transmission delay for the Cas-MAC protocol finally becomes

$$D_{\text{Cas-MAC}} = D_{\text{Cas-MAC}_{\text{1st hop}}} + (N_{\mathrm{hop}} - 1)\xi. \tag{15}$$

5.3. Experimental validation

We now validate experimentally the analytical latency models corresponding to the two considered MAC protocols. The Cas-MAC protocol has been implemented in TinyOS-2.1 for MicaZ motes, by modifying the original SCP-MAC protocol implementation provided by the UPMA/MLA framework [22]. While using the same sending routine, we have modified the synchronization policy to enforce duty cycles skewing through explicit synchronization phases initiated by the sink node at regular intervals. We have also used the X-MAC protocol implementation available in the UPMA/MLA framework to perform benchmarking with our Cas-MAC implementation.

In order to evaluate the implementation correctness of both X-MAC and Cas-MAC protocols, we use Avrora, a cycle-accurate instruction-level simulation and analysis framework for the sensor networks, built on the Atmel

[7] Note that the symbol d is also used to refer to the distance between two nodes. The context eliminates any ambiguity.

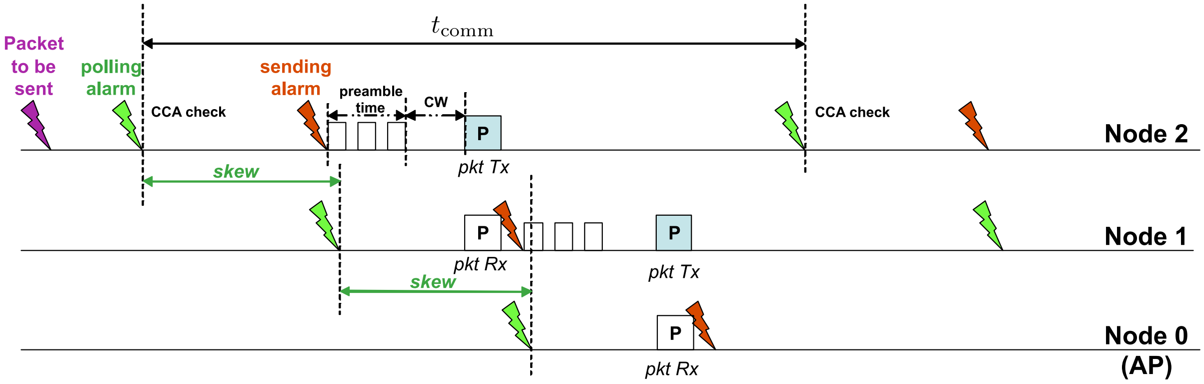


Fig. 8. Design philosophy of the Cas-MAC protocol.

AVR micro controller, presented in [27]. The Avrora framework allows the precise emulation of IEEE 802.15.4-based protocols without any modifications in the code developed for the real hardware. In [28], the correctness of the emulation of the Texas Instruments Chipcon CC2420 radio chip, used in many sensor node platforms (such as the Crossbow MicaZ nodes), is proved.

For each of the experiments presented hereafter, the average latency is evaluated considering 100 sample packets traversing a 6-hop chain path. For the X-MAC protocol, we observed that sending packets leads to aligned duty cycles, thus biasing the latency measurements. In order to eliminate this unintended effect, we decided to run one simulation for each single sample packet and to average over the set of simulations.

In Fig. 9a and b, the multi-hop latencies with the X-MAC and Cas-MAC protocols, respectively, are shown as functions of the number of hops. As can be seen from the obtained results, there is an overall good agreement between experimental and analytical results. The average latency increases linearly as a function of the number of a hops with a slope approximating $t_{\text{comm}}/2$ for the X-MAC protocol and ξ for the Cas-MAC protocol.

In order to evidence the low latency guaranteed by the Cas-MAC protocol, in Fig. 9c we evaluate the 6-hop latency for various values of the duty cycle, considering both X-MAC and Cas-MAC protocols. In order to carry out this comparison, we fix the portion of time during which the node is active, and increase the sleep interval, thus increasing the total duration t_{comm} of the duty cycle. For very low values of β_{comm}, when the node is sleeping during a large fraction of time, the Cas-MAC protocol outperforms the X-MAC protocol. However, when β_{comm} gets higher, i.e., the node is active for a larger fraction of time, the benefits of the Cas-MAC protocol disappear.

To properly tune the Cas-MAC protocol, a key factor is the length of the skew, which indicates the offset on each hop along the path. If ξ is shortened, the average latency, which increases linearly as a function of number of hops with slope ξ, reduces. However, it becomes more likely that the sending alarm of a node will fire before it has received the packet. In the latter case, this intermediate node will have to buffer the packet until the next sending opportunity, which will result in increasing the average multi-hop latency. In Fig. 9d, we show the 6-hop latency with the Cas-MAC protocol, using multiple values of ξ. The presented results underline that there is a threshold value of ξ below which the average latency increases. This helps to determine the margin required because of clock drift and processing time fluctuations.

6. A cross-layer network lifetime model

We now propose a simple energy model to complete our modeling framework. Node energy is a crucial common denominator in our cross-layer modeling, as all the critical functions (at sensing and communication layers) contribute to its depletion and, thus, to the decrease of system lifetime. The energy model creates the inter-dependencies between all functional layers that guide the performance trade-offs that a WSN operator may face between reliability, reactivity, and sustainability.

The energy consumed by a node is defined as the sum of the energies consumed by its hardware components. For the sake of simplicity, we only integrate in the energy model the contributions from the sensing sub-unit and the radio transceiver. The main parameters of the energy model are listed in Table 3.

The network lifetime is defined as the time needed for the average residual energy in the deployed WSN, denoted E_{r}, to become lower than a threshold value E_{th}. Denoting the initial energy of a node as E_{i} and denoting the power consumption (due to sensing and communication operations) of the whole network (i.e., at all nodes) as Ω_{cons}, the average residual energy E_{r} at time t (assuming that the network turns on at time 0) can be expressed as

$$E_{\text{r}}(t) = NE_{\text{i}} - \Omega_{\text{cons}}t. \tag{16}$$

We now focus on the derivation of the energy model for the Cas-MAC protocol, as the energy model for the X-MAC protocol can be found in [6]. According to the description of the Cas-MAC protocol in Section 5, Ω_{cons} can then be computed as follows:

$$\Omega_{\text{cons}} = [(\Omega_{\text{R}} + \Omega_{\text{T}})P_{\text{d}}N_{\text{target}} + \Omega_{\text{LPL}}]\left(1 - \frac{t_{\text{comm}}}{t_{\text{sync}}}\right) + \Omega_{\text{sync}}\frac{t_{\text{comm}}}{t_{\text{sync}}} + \Omega_{\text{sensing}}, \tag{17}$$

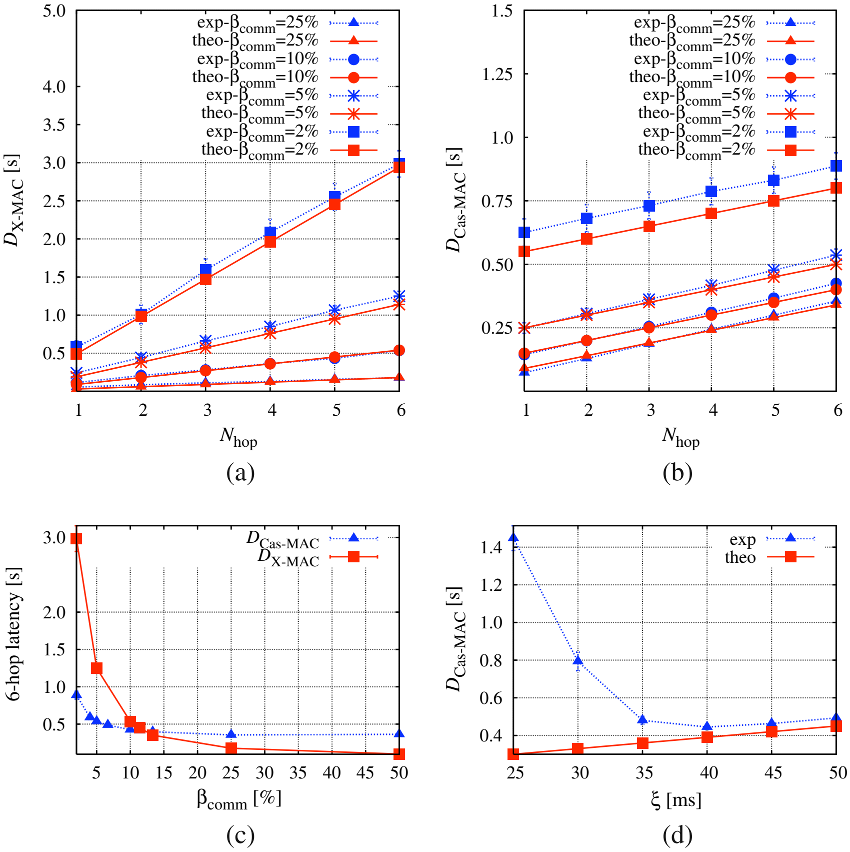


Fig. 9. Experimental results on latency measurements: (a) validation of X-MAC's latency model, with experimental (dashed lines) against theoretical (solid lines) results; (b) validation of Cas-MAC's latency model, with experimental (dashed lines) against theoretical (solid lines) results; (c) comparison of the 6-hop experimental latency for X-MAC (solid lines) and Cas-MAC (dashed lines) as function of β_{comm}; and (d) theoretical (solid lines) and experimental (dashed lines) latency results as function of the skew ξ, with $N_{\text{hop}} = 6$ and $\beta_{\text{comm}} = 5\%$.

Table 3
Constant and variable system parameters considered in the energy-model.

Constant parameters			
Number of nodes in the network	N	25	Scenario-specific
Preamble duration	S_p	0.26 ms	Configuration-specific
Packet duration	S_d	0.93 ms	Application-specific
Contention window duration	CW	5 ms	Configuration-specific
Wake-up tone length	WT_{length}	20 ms	Configuration-specific
Transmission power consumption	Ω_{T_x}	0.0511 W	Device-specific
Reception power consumption	Ω_{R_x}	0.0588 W	Device-specific
Sensing power consumption	Ω_{sens}	0.0036 W	Device-specific
Sleep power consumption	Ω_{sleep}	2.4×10^{-7} W	Device-specific
Variable parameters (with default value ranges)			
Average number of hops	N_{hop}	3.0, 3.1, 4.0, 5.8	Function of the topology
Sensing duty cycle	β_{sens}	0.1–1	Configuration-specific
Sensing period	t_{sens}	5–25 (15) s	Configuration-specific
Communication duty cycle	β_{comm}	0.002–1	Configuration-specific
Communication period	t_{comm}	40–1000 (100) ms	Configuration-specific
Synchronization period	t_{sync}	30 s	Configuration-specific
Number of incoming targets per day	N_T	10	Scenario-specific

where N_{target} is the number of target appearances, P_d is the target detection probability, t_{comm} is the communication period, and t_{sync} is the synchronization interval. One can distinguish the following sources of power consumption:

- Ω_T and Ω_R are the powers used to send and receive an alert message over a period of duration t_{comm}, respectively;
- Ω_{sync} is the power used for synchronization operations over a period of duration t_{sync};

- Ω_{LPL} is the power required when performing the Low Power Listening (LPL) operations over a period of duration t_{comm};
- Ω_{sensing} is the power consumption associated with the sensing module in the activity period of duration t_{sens}.

The expected power Ω_{T} to send a data packet can be expressed as

$$\Omega_{\text{T}} = [\Omega_{\text{T}_\text{x}}(WT_{\text{length}} + S_{\text{d}}) + \Omega_{\text{R}_\text{x}} CW] \frac{N_{\text{hop}}}{t_{\text{comm}}}, \tag{18}$$

where $\Omega_{\text{T}_\text{x}}(WT_{\text{length}} + S_{\text{d}})$ is the energy spent to transmit a wake-up tone followed by a data packet and $\Omega_{\text{R}_\text{x}} CW$ is the energy spent while sensing the channel in order to detect potential transmissions from other neighboring nodes. The presence of the multiplicative term N_{hop} takes into account the average number of relays from the sensor node to the sink. The value of N_{hop} can be determined by considering all possible shortest (in terms of number of hops) paths using the Dijkstra algorithm [29].

Similarly, the expected power Ω_{R} to receive a data packet can be expressed as

$$\Omega_{\text{R}} = \Omega_{\text{R}_\text{x}}(S_{\text{d}} + WT_{\text{length}}) \frac{N_{\text{hop}}}{t_{\text{comm}}}. \tag{19}$$

The power consumed during synchronization operations can be expressed as

$$\Omega_{\text{sync}} = \Omega_{\text{Tx}_{\text{sync}}} + (N-1)\Omega_{\text{Rx}_{\text{sync}}}, \tag{20}$$

where $\Omega_{\text{Tx}_{\text{sync}}}$ and $\Omega_{\text{Rx}_{\text{sync}}}$ are the powers consumed for transmitting and receiving the synchronization message, respectively. In particular, the power $\Omega_{\text{Tx}_{\text{sync}}}$, consumed by the sink for the transmission of the sync packet, can be given by the following expression:

$$\Omega_{\text{Tx}_{\text{sync}}} = \frac{S_{\text{d}} \Omega_{\text{Tx}}}{\frac{3}{4}\beta_{\text{comm}} t_{\text{sync}}} \tag{21}$$

where t_{sync} is the time interval occurring between two synchronizations and the fact that the sink transmits, every $\frac{3}{4}\beta_{\text{comm}} t_{\text{comm}}$, a sync packet is taken into account. Similarly, the power $\Omega_{\text{Rx}_{\text{sync}}}$, consumed by the remote nodes to receive the synchronization packet, can be expressed as

$$\Omega_{\text{Rx}_{\text{sync}}} = \frac{S_{\text{d}} \Omega_{\text{Rx}}}{\frac{3}{4}\beta_{\text{comm}} t_{\text{sync}}}. \tag{22}$$

The power associated with the LPL, or duty cycling, operations can be expressed as

$$\Omega_{\text{LPL}} = N\Omega_{\text{R}_\text{x}}\beta_{\text{comm}} + N\Omega_{\text{sleep}}(1 - \beta_{\text{comm}}) - \Gamma_{\text{T}_\text{x}} \tag{23}$$

where $\Gamma_{\text{T}_\text{x}}$ is a correction term used to refine the power consumption due to LPL operations. In fact, the transmission operations overlap, for a short interval, with the LPL operations. Therefore, without the correction term $\Gamma_{\text{T}_\text{x}}$, the power consumption budget would be higher than the correct one. In particular, $\Gamma_{\text{T}_\text{x}}$ can be expressed as

$$\Gamma_{\text{T}_\text{x}} = (WT_{\text{length}} + CW + S_{\text{d}})\Omega_{\text{sleep}} \frac{N_{\text{hop}}}{t_{\text{comm}}}.$$

In fact, term $\Gamma_{\text{T}_\text{x}}$ takes into account that during transmission operations, such as (i) preamble transmission, (ii) contention window, and (iii) packet transmission, the node would normally be in the sleep state.

Finally, the power consumed during sensing operations can be expressed as

$$\Omega_{\text{sensing}} = \beta_{\text{sens}} \Omega_{\text{sens}} N. \tag{24}$$

Note that in the case of the X-MAC protocol, the derivation is slightly different for Ω_{LPL}, Ω_{R} and Ω_{T} (see [6] for more details), since the X-MAC protocol does not have any synchronization mechanism. In addition, while with the Cas-MAC protocol a node with data broadcasts a wake-up tone constituted by short preambles and then transmits the data packet without acknowledgment, the X-MAC protocol opts for the preambling of the very data packet and its acknowledgment by the receiving node.

Finally, the network lifetime L can be defined as the interval between the initial instant at which the nodes are turned on and the time instant at which the residual network energy $E_r(L)$ becomes equal to NE_{th}, where E_{th} is a given residual node energy threshold (which takes into account the physical behavior of the device). From (16), the following expression for the network lifetime L can be straightforwardly derived:

$$L = \frac{N(E_{\text{i}} - E_{\text{th}})}{\Omega_{\text{cons}}}. \tag{25}$$

7. System engineering: Two application cases

In this section, we consider two illustrative application cases to show how the cross-layer modeling framework can help a network operator to predict the performance of a deployed WSN. We first show how to optimize the operational system parameters for a given topology. Then, we highlight the impact of nodes' placement on the expected performance.

7.1. Maximizing the system lifetime for given (illustrative) topologies

In this subsection, we present an application case in which a given surveillance system is optimally configured to maximize the network lifetime L, for given Quality of Service (QoS) requirements in terms of probability of missed detection (P_{md}) and latency after detection (D). Throughout this subsection, the target arrival rate N_{target} is set to 10 targets per day. In Fig. 10, the considered topology, with 25 nodes deployed over a 1 km × 1 km area and with the sink located at the bottom left corner of the monitored area, is shown. The considered network is completely connected, the average number of hops is 3, and the lowest achievable (with $\beta_{\text{sens}} = 1$) value of P_{md} is 0.6.

This application case consists of the optimization of a single-objective function (the network lifetime), given constraints on the two other functions (the maximum tolerable probability of missed detection, denoted as P^*_{md}, and the longest tolerable latency after detection, denoted as D^*). Since Eqs. (5) and (25) are not linear, standard linear programming optimization techniques cannot be used. However, the three equations identify a convex set, which

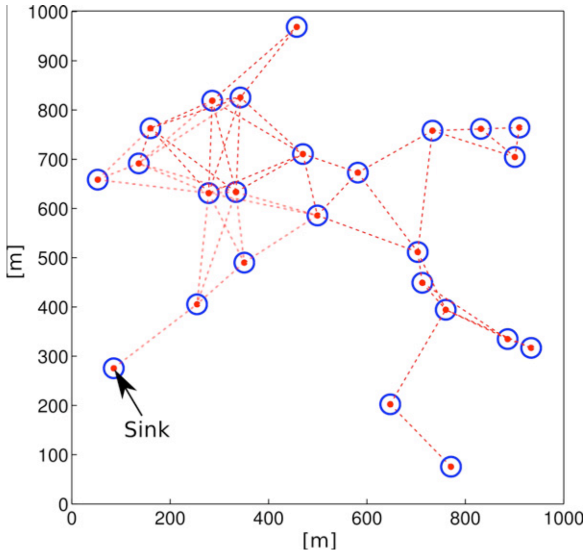


Fig. 10. Illustrative example of a connected topology.

makes gradient-based optimization techniques feasible [30].

In Fig. 11, the isolines corresponding to the maximum lifetimes, with both the X-MAC and Cas-MAC protocols, are shown as functions of the maximum tolerable values $P^*_{\rm md}$ and D^*—the underlying constraints for lifetime maximization are $P_{\rm md} < P^*_{\rm md}$ and $D < D^*$. The trends of the isolines confirm the trade-offs that a network operator faces when deploying and configuring a WSN. For any reachable value of maximum lifetime, there exists a multitude of operational points that give a varying order of precedence to the set of constrained performance indicators. For instance, when using the X-MAC protocol, the operator knows that the maximum lifetime of 200 days is reachable for various QoS conditions (e.g., $P_{\rm md} < 85$ % and $D < 2$ s, or $P_{\rm md} < 75$ % and $D < 4$ s). This highlights the cross-layer nature of our optimization, where the energy model is a common denominator for the operations of both sensing and communication layers. More precisely, for a given energy consumption, the performance at one layer can be improved at the cost of a performance degradation at the other layer.

Considering any of the MAC protocols and a given constraint on $P_{\rm md}$ (D, respectively), the maximum lifetime is shorter when making the requirement on D ($P_{\rm md}$, respectively) more stringent. This is due to the fact that a sensor must keep its communication (sensing, respectively) interface on for a larger portion of each communication (sensing, respectively) duty cycle, so that the consequent energy consumption increases and the network lifetime shortens.

Last, but not least, we can also assess the influence of the MAC protocol on the reachable levels of QoS. The contour maps of $L_{\rm max} = 100$ and $L_{\rm max} = 200$ show that a given lifetime can be reached with more constrained operational points, in terms of $P_{\rm md}$ and D, when using the Cas-MAC protocol rather than the X-MAC protocol. In other words, the Cas-MAC protocol broadly guarantees a better QoS than the X-MAC protocol.

7.2. Impact of node placement

The influence of the topology on the predicted performance is another cross-layer aspect of our modeling framework. The placement of nodes, indeed, has an impact on the average number of communication hops to the sink, related to the latency after detection, as well as the delay before detection and the probability of missed detection. In other words, the placement influences each QoS indicator of the target detection-oriented WSN-based system.

In Fig. 12, optimization results, relative to the three placement strategies illustrated in Fig. 13, are shown. Each topology is associated with the specific values of $N_{\rm hop}$ and $P_{\rm md}$ (obtained for $\beta_{\rm sens} = 1$) shown in Table 4. In all cases, the sink is the node at the bottom left of the area. All of the topologies are connected, in the sense that every node

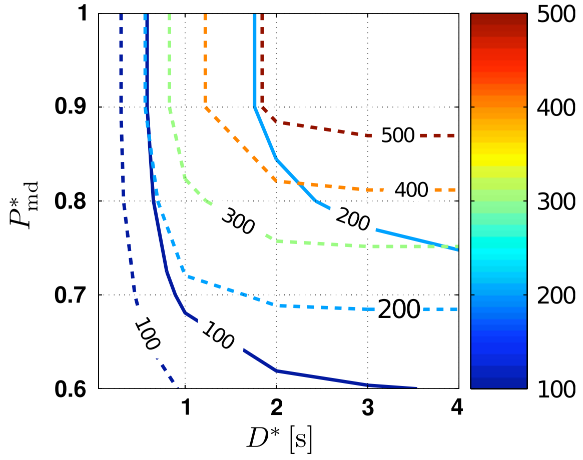


Fig. 11. Contour lines of maximum lifetime (in days) as functions of constraints on $P_{\rm md}$ and D (i.e., their maximum tolerable values $P^*_{\rm md}$ and D^*), when using the X-MAC (solid lines) and Cas-MAC (dash-dot lines) protocols.

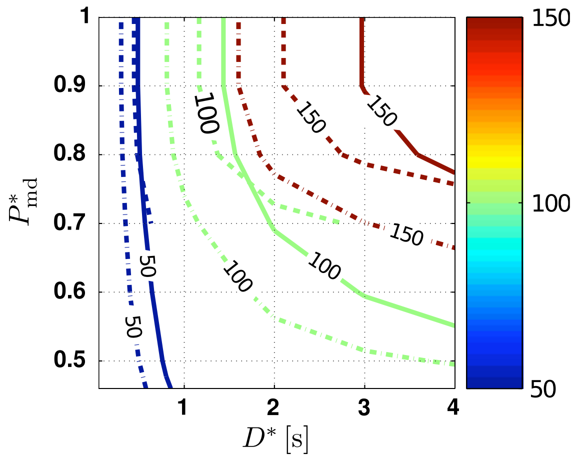
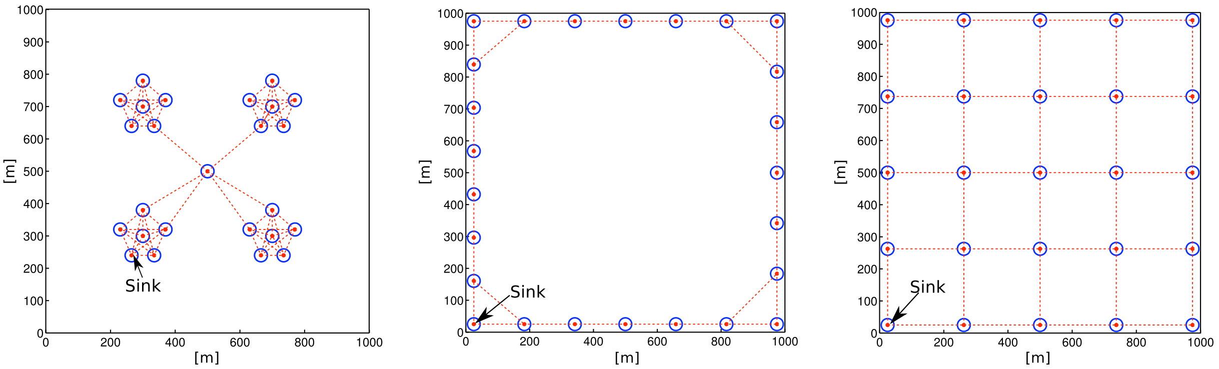


Fig. 12. Contour lines of maximum lifetime (in days) as functions of constraints on $P_{\rm md}$ and D (i.e., their maximum tolerable values $P^*_{\rm md}$ and D^*), when using the X-MAC protocol, for grid (dash-dot lines), border (solid lines), and cluster (dashed lines) topologies.



(a) (b) (c)

Fig. 13. Examples of node placement strategies: (a) cluster, (b) border, and (c) grid topology.

Table 4
N_{hop} and best P_{md} associated to the topologies presented in Fig. 13.

Topology	N_{hop}	Best value of P_{md}
Cluster	3.1	0.71
Grid	4.0	0.46
Border	5.8	0.45

is within the radio coverage of at least one neighbor, and a multi-hop path exists from each node to the sink. For each topology, we have represented a set of isolines for the maximum lifetime as function of the constraints $P_{\text{md}} < P^*_{\text{md}}$ and $D < D^*$, when using the X-MAC protocol.

The set of plots provides a number of insights on the selection of the placement strategy.

- *The topology has a direct influence on the "landscape" of achievable operational points*. For instance, when using the *cluster* topology, the best reachable value of P_{md} is 0.7. The bottom line is that the optimization technique cannot find any (maximum) value of the lifetime under the constraint $P_{\text{md}} < P^*_{\text{md}}$ with $P^*_{\text{md}} = 0.7$. In the case that an operator requires to operate the system with more demanding QoS in terms of detection capability (i.e., $P_{\text{md}} < 0.7$), he should opt for the *grid* topology or the *border* topology.
- *The average number of communication hops is a key factor for QoS provisioning*. Considering the *grid* and *border* topologies (i.e., with a respective average number of hops of 4.0 for the *grid*, and 5.8 for the *border*), for a given constraint on the delay (e.g., $D < D^*$ with $D^* = 4$ s), the maximum lifetime of 100 days is obtained with a better QoS in terms of P_{md} when using the *grid* topology (i.e., P_{md} smaller than 0.49), rather than the *border* topology (P_{md} smaller than 0.55). This pertains to the operations of the X-MAC protocol, as the model for the delay shows that the latency depends on the number of hops and on the value of the duty cycle. In particular, for a given constraint on the delay, the larger the number of hops, the lower the one-hop delay, and the longer the duty cycle, thus increasing the energy budget for the communication layer. As a consequence, the same value of maximum lifetime can be reached only by relaxing the energy budget dedicated to the sensing layer, i.e., by relaxing the constraint on P_{md}.

Note that, in the case of the Cas-MAC protocol, the impact of the node placement, in terms of average number of hops to the sink, is reduced with respect to that required by the X-MAC protocol. Indeed, the latency model shows a dependence on the communication duty cycle only for the first hop. For the following hops, the one-hop delay only depends on the introduced offset, regardless of the duty cycle. Since, in the lifetime model, the energy budget for communication only takes into consideration the average number of hops, the isolines tend to be very close, as shown in Fig. 14. In particular, considering the *grid* and *border* topologies (i.e., with corresponding average number of hops equal to 4.0 and 5.8, respectively), the *grid* topology allows to reach operational points slightly more constrained than those allowed by the *border* topology. Comparing the *cluster* topology with the *grid* and *border* topologies, one can see that the most constrained isolines

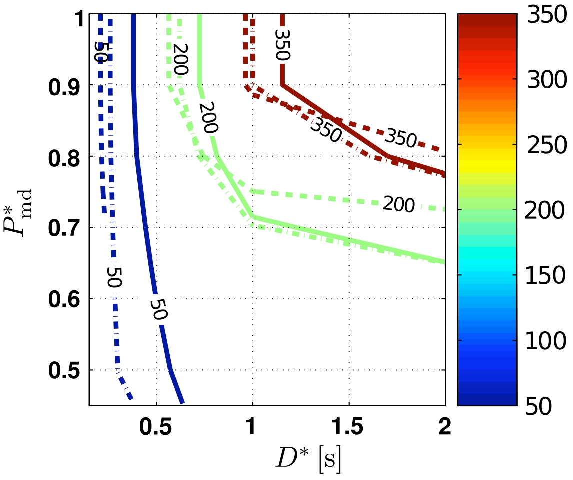


Fig. 14. Contour lines of maximum lifetime (in days) as functions of constraints on P_{md} and D (i.e., their maximum tolerable values P^*_{md} and D^*), when using the Cas-MAC protocol, for grid (dash-dot lines), border (solid lines), and cluster (dashed lines) topologies.

(e.g., maximum lifetime equal to 50 days) are achieved when using the *cluster* topology, rather than with the *grid* and *border* topologies. On the opposite, one can see that the least constrained isolines (e.g., maximum lifetime equal to 200 days or 350 days) are achieved for more constrained operational points when using the *grid* and *border* topologies, rather than with the *cluster* topology, especially for loose constraints on D. In that case, the energy budget due to the operations of the MAC protocol tends to become very close for all the topologies, and the budget portion owing to sensing unit weighs more, at the disadvantage of the *cluster* topology.

To summarize, a few illustrative examples have shown that the node placement can have a relevant impact in terms of both achievable operational points and QoS provisioning.

8. Concluding remarks

This paper has addressed the problem of engineering energy-efficient mobile target detection applications using WSNs with deterministic (a priori known) node deployment. In particular, we have first proposed an analytical framework for the evaluation of several performance metrics at sensing and communication layers: the probability of missed detection, the delay before the first detection act, the latency after detection, and the energy consumption. We have then characterized, using this toolbox, the cross-layer interactions between sensing and communication layers, evaluating the energy consumption under given constraints in terms of detection capability and latency. Finally, we have illustrated the use of our toolbox to predict the performance of practical WSN-based surveillance systems under specific QoS requirements. Our results show clearly that the network topology and the MAC protocol have an impact on sensing and communication system capabilities and, therefore, lead to cross-layer tradeoffs. For instance, the novel Cas-MAC protocol guarantees a threefold network lifetime extension with respect to the X-MAC protocol, considering given QoS requirements on the detection capability and responsiveness of the WSN. Focusing on illustrative relevant network topologies, we have shown that the system can reach more constrained operational points, in terms of probability of missed detection, using specific node placements (e.g., *border* or *grid* topologies rather then *cluster* topologies).

Further work along these lines include the extension of the engineering toolbox to take into account the use of complex nodes, in order to improve the overall accuracy of the analytical performance predictions. In particular, there is room to refine the cross-layer models in order to encompass the effect of more realistic radio and sensing environments. At last, we envision to assess the validity of the predicted performance through large-scale field testing with real hardware platforms.

Appendix A. Geometric derivation of $m_2(i,j)$

The term $m_2(i,j)$ can be evaluated, as shown in [20], as a function of the distance between the sensor nodes i and j [31]. In particular, under the assumption of equal sensing ranges, $m_2(i,j)$ can be expressed as follows:

$$m_2(i,j) = \begin{cases} 2\pi r_s + 2\pi r_s - L_{out}(d_{i,j}) & \mathscr{A}_i \cap \mathscr{A}_j \neq 0 \\ L_{in}(d_{i,j}) - L_{out}(d_{i,j}) & \mathscr{A}_i \cap \mathscr{A}_j = 0 \end{cases}, \qquad \text{(A.1)}$$

where $d_{i,j}$ is the distance between the sensor nodes i and j, $L_{in}(d_{i,j})$ ($L_{out}(d_{i,j})$, respectively) denotes the length of the inner (outer, respectively) string wrapped around the sensing areas of nodes i and j, as shown in Fig. A.15.

After a few geometrical considerations, one can show that

$$L_{out}(d_{i,j}) = 2\pi r_s + 2d_{i,j},$$

$$L_{in}(d_{i,j}) = 2r_s\left[2\pi - 2\arccos\left(\frac{2r_s}{d_{i,j}}\right)\right] + 4\sqrt{\frac{d_{i,j}^2}{4} - r_s^2}.$$

Appendix B. Derivation of $P\{\mathcal{E}_{det}|\mathcal{E}_{SoT}\}$

Since the target moves at constant speed v, the crossing time of a sensed area is $T_{cross} = L/v$, where L is the length of the intersection between the target trajectory and the area sensed by a sensor, as shown in Fig. B.16. As L is a random variable distributed in $[0, 2r_s]$, T_{cross} is a random variable with distribution in $[0, 2r_s/v]$. Since there is no information about the arrival instant of the target *and* under the assumption of infinite duty cycling, the delay with respect to beginning of the last duty cycle can be modeled as a random variable uniformly distributed in $[0, t_{sens}]$. Without loss of generality, we now focus on a single period of duration t_{sens}. When the sensor is on, i.e., during the subinterval of duration $\beta_{sens}t_{sens}$, any incoming target will be detected. In the case that the sensor is off, i.e., during the subinterval of duration $(1-\beta_{sens})t_{sens}$, the following analysis can be carried out. Let $\mathscr{E}_{target}$ be the event {The sensor is on at the instant at which the target enters the sensed area}. Applying the total probability theorem [32], $P\{\mathscr{E}_{det}|\mathscr{E}_{SoT}\}$ can then be expressed as

$$\begin{aligned} P\{\mathscr{E}_{det}|\mathscr{E}_{SoT}\} &= P\{\mathscr{E}_{det}|\mathscr{E}_{target}, \mathscr{E}_{SoT}\}P\{\mathscr{E}_{target}|\mathscr{E}_{SoT}\} \\ &\quad + P\{\mathscr{E}_{det}|\overline{\mathscr{E}}_{target}, \mathscr{E}_{SoT}\}P\{\overline{\mathscr{E}}_{target}|\mathscr{E}_{SoT}\}, \end{aligned} \qquad \text{(B.1)}$$

where it is immediate to conclude that $P\{\mathscr{E}_{det}|\mathscr{E}_{target}, \mathscr{E}_{SoT}\} = 1, P\{\mathscr{E}_{target}|\mathscr{E}_{SoT}\} = \beta_{sens}$, and $P\{\overline{\mathscr{E}}_{target}|\mathscr{E}_{SoT}\} = 1 - \beta_{sens}$. Therefore, Eq. (B.1) can be rewritten as:

$$P\{\mathscr{E}_{det}|\mathscr{E}_{SoT}\} = \beta_{sens} + (1-\beta_{sens})P\{\mathscr{E}_{det}|\overline{\mathscr{E}}_{target}, \mathscr{E}_{SoT}\}. \qquad \text{(B.2)}$$

The conditional probability at the right-hand side of (B.2) can be expressed as [6]

$$P\{\mathscr{E}_{det}|\overline{\mathscr{E}}_{target}, \mathscr{E}_{SoT}\} = \int\int_{\mathscr{D}} f_{T_a,T_{cross}}(t,\tau)\,dt\,d\tau,$$

where the two-dimensional domain $\mathscr{D}$ can be expressed as follows:

$$\mathscr{D} = \{(t,\tau) \in \mathbb{R}^2 : 0 < \tau < 2r_s/v \quad \text{and}$$
$$\max\{-\tau + c, 0\} < t < c\},$$

where $c \triangleq (1-\beta_{sens})t_{sens}$. An illustrative representation of $\mathscr{D}$ is shown in Fig. B.17, distinguishing two cases: (a)

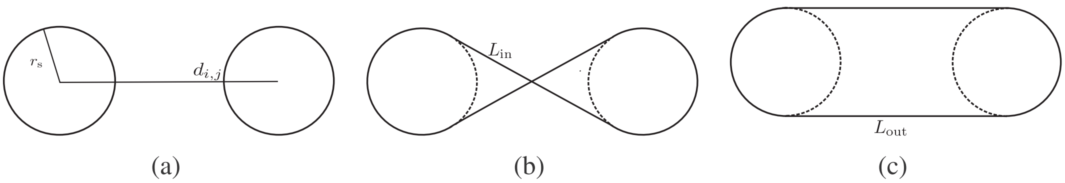


Fig. A.15. Graphical derivation of L_{in} and L_{out} for a pair of nodes with sensing range r_{s} at a distance $d_{i,j}$: (a) reference scenario, (b) closed line of length $L_{\mathrm{in}}(d_{i,j})$, and (c) closed line of length $L_{\mathrm{out}}(d_{i,j})$.

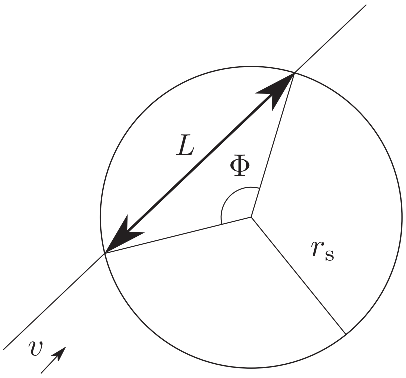


Fig. B.16. Model for the characterization of the intersection of a target trajectory with a sensed area.

$2r_{\mathrm{s}}/v > c$ and $2r_{\mathrm{s}}/v < c$. The joint probability density function (pdf) $f_{T_{\mathrm{a}},T_{\mathrm{cross}}}(t,\tau)$ can be expressed as [6]:

$$f_{T_{\mathrm{a}},T_{\mathrm{cross}}}(t,\tau) = \begin{cases} \frac{v}{\pi c\sqrt{r_{\mathrm{s}}^2-\left(\frac{v\tau}{2}\right)^2}} \\ \quad 0 < \tau < 2r_{\mathrm{s}}/v \quad \text{and} \quad 0 < t < c \\ 0 \quad \text{else}. \end{cases} \tag{B.3}$$

At this point, given that $T_{\mathrm{a}} \sim \mathrm{Unif}\,[0,c]$, it can be shown that

$$P\{\mathscr{E}_{\mathrm{det}}|\overline{\mathscr{E}}_{\mathrm{target}},\mathscr{E}_{\mathrm{SoT}}\} = \begin{cases} \frac{4r_{\mathrm{s}}}{\pi c v} \\ \quad \text{if } 2r_{\mathrm{s}}/v < c \\ \frac{4r_{\mathrm{s}}-2\sqrt{4r_{\mathrm{s}}^2-c^2v^2}}{\pi c v} + 1 - \frac{2\mathrm{asin}\left(\frac{cv}{2r_{\mathrm{s}}}\right)}{\pi} \\ \quad \text{otherwise}. \end{cases} \tag{B.4}$$

Appendix C. Derivation of $P\left\{\mathcal{E}_{\mathbf{det}}^{(\boldsymbol{i,j})}|\mathcal{E}_{\mathbf{SoT}_{\boldsymbol{i,j}}}\right\}$

$P\left\{\mathscr{E}_{\mathrm{det}}^{(i,j)}|\mathscr{E}_{\mathrm{SoT}_{i,j}}\right\}$ is the probability that the target is detected during the active phase of at least one of the two sensors i and j, *given that* the target trajectory crosses both sensed areas. Since the target trajectory is random and there is no synchronization between the two sensors, and since the target detection can be carried out by (i) only node i, (ii) only node j, or (iii) both nodes, $P\left\{\mathscr{E}_{\mathrm{det}}^{(i,j)}|\mathscr{E}_{\mathrm{SoT}_{i,j}}\right\}$ can be expressed as

$$\begin{aligned} P\left\{\mathscr{E}_{\mathrm{det}}^{(i,j)}|\mathscr{E}_{\mathrm{SoT}_{i,j}}\right\} &= P\left\{\mathscr{E}_{\mathrm{det}}^{(i)}|\mathscr{E}_{\mathrm{SoT}_{i,j}}\right\} + P\left\{\mathscr{E}_{\mathrm{det}}^{(j)}|\mathscr{E}_{\mathrm{SoT}_{i,j}}\right\} \\ &\quad - P\left\{\mathscr{E}_{\mathrm{det}}^{(i)}|\mathscr{E}_{\mathrm{SoT}_{i,j}}\right\} \cdot P\left\{\mathscr{E}_{\mathrm{det}}^{(j)}|\mathscr{E}_{\mathrm{SoT}_{i,j}}\right\}, \end{aligned} \tag{C.1}$$

where $P\left\{\mathscr{E}_{\mathrm{det}}^{(i)}|\mathscr{E}_{\mathrm{SoT}_{i,j}}\right\}$ is the probability that node i is active, given that the target crosses its sensed area (and that of node j), and $P\left\{\mathscr{E}_{\mathrm{det}}^{(j)}|\mathscr{E}_{\mathrm{SoT}_{i,j}}\right\}$ is the probability that node j is active, given that the target crosses its sensed area (and that of node i).

The allowed trajectories for the computation of m_2 (i.e., the trajectories which belong to both sensed areas) are a subset of all admissible ones for a single sensor, as shown, in terms of possible positions of the entrance point over the perimeter of a sensed area, in Fig. C.18a. In fact, in the case of m_1, the entrance point of the target in the area sensed by a sensor can be the whole perimeter of the sensed area, so that the angle Φ, shown in Fig. B.16, can range in the interval $[0,2\pi]$. Conversely, in the m_2 case, one must consider, for node i, only the entrance points which allow the target to cross both sensed areas. Referring to the logical scheme shown in Fig. C.18a, the range of allowed entrance points for the first sensed area can

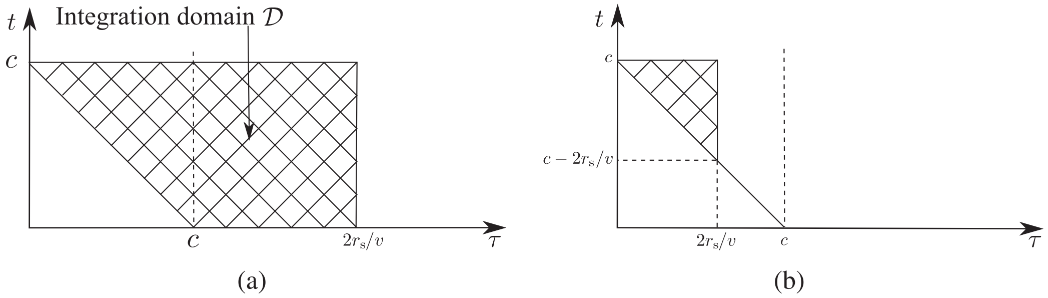


Fig. B.17. Integration domain for the evaluation of $P\{\mathscr{E}_{\mathrm{det}}|\overline{\mathscr{E}}_{\mathrm{target}},\mathscr{E}_{\mathrm{SoT}}\}$: (a) $2r_{\mathrm{s}}/v > c$ and (b) $2r_{\mathrm{s}}/v < c$.

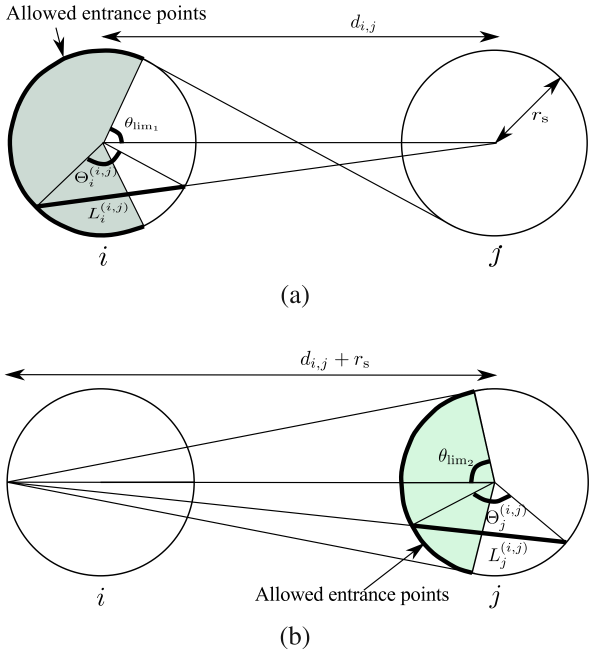


Fig. C.18. Allowed entrance points: (a) along the area sensed by node *i*, to ensure that the target crosses also the sensed area of node *j*, and (b) along the area sensed by node *j*, to ensure that the target crosses also the sensed area of node *i*.

be approximated as the arc, highlighted in Fig. C.18a, which insists on the angle $(2\pi - 2\theta_{\lim_1})$, where $\theta_{\lim_1}$ is a function of $d_{i,j}$ (the distance between sensor i and sensor j) and r_s. For each admissible entrance point position, the entrance angle belongs to a specific (limited) interval.

In order to simplify the derivation of $P\left\{\mathscr{E}_{\text{det}}^{(i)}|\mathscr{E}_{\text{SoT}_{i,j}}\right\}$, we must find the distribution of the length $L_i^{(i,j)}$ of the intersection between the target trajectory and the area sensed by node i. Moreover, one also needs to compute, for node i, the joint pdf $f_{T_a,T_{\text{cross}}^{(i)}}(t,\tau)$, which replaces that presented in expression (B.3). Similar considerations can be carried out for the derivation of $P\left\{\mathscr{E}_{\text{det}}^{(i)}|\mathscr{E}_{\text{SoT}_{i,j}}\right\}$ which, as shown in Fig. C.18b, depends on the entrance point of the node in the area sensed by node j. Assuming that the target has already crossed the area sensed by node i, the range of possible entrance points lies on the arc, highlighted in Fig. C.18b, which insists on the angle $(2\pi - 2\theta_{\lim_2})$. As for node i, we simplify the derivation of the distribution of the length $L_j^{(i,j)}$ of the intersection between the target trajectory and the area sensed by j. In particular, referring to Fig. C.18b, we assume that all targets enter in the area sensed by node i in the point lying on the line connecting the nodes i and j.

On the basis of geometrical considerations—omitted here for conciseness—the length of a chord L can be expressed as the following function of the angle Θ[8]:

Table C.5
Coefficients for the polynomial approximation of (C.4).

Coefficients	Analytical expressions	Numerical values
a_1	$-\frac{2r_s}{(\pi-\theta_{\lim})^2}$	−38.06
a_2	$-2a_1\pi$	239.17
a_3	$a_1\pi^2 + 2r_s$	−275.69

$$L = B_1 - \left\{ r_s^2 + d^2 - 2r_s d \left[-\left\{ \left(1 - \frac{d\ \sin\Theta}{B_1}\right) \cdot \left(1 - \frac{r_s\ \sin\Theta}{B_1}\right) \right\}^{1/2} + \frac{r_s d \sin^2\Theta}{B_1^2} \right] \right\}^{1/2} \tag{C.2}$$

where $B_1 \triangleq \sqrt{r_s^2 + d^2 - 2r_s d\cos\Theta}$, and

$$d = \begin{cases} d_{i,j} & \text{if } L = L_i^{(i,j)} \\ d_{i,j} + r_s & \text{if } L = L_j^{(i,j)}. \end{cases} \tag{C.3}$$

Since expression (C.2) is complicated, it can be shown that it can be accurately approximated through a degree-2 polynomial approximation.

More precisely, assuming that

$$L \simeq a_1\Theta^2 + a_2\Theta + a_3 \tag{C.4}$$

where a_1, a_2, and a_3 are shown in Table C.5, it is possible to evaluate the joint pdf which replaces that of Eq. (B.3).[9] Eq. (C.2) and its approximation (C.4) should depend on the distance between the centers of nodes i and j. However, after some algebraic manipulations, the dependence on the distance between the pair of sensors disappears and it can be shown that:

$$f_{T_a,T_{\text{cross}}}^{(i)}(t,\tau) = f_{T_a,T_{\text{cross}}}^{(j)}(t,\tau) = \begin{cases} \frac{v}{c\sqrt{16r_s^2 - 8r_s v\tau}} & 0 < \tau < 2r_s/v \quad \text{and} \quad 0 < t < c \\ 0 & \text{else.} \end{cases} \tag{C.5}$$

The evaluation of $P\left\{\mathscr{E}_{\text{det}}^{(i)}|\mathscr{E}_{\text{SoT}_{i,j}}\right\}$ depends on the relation between $2r_s/v$ and $(1-\beta_{\text{sens}})t_{\text{sens}}$, but not on node i. In particular, it can be shown that

$$P\{\mathscr{E}_{\text{det}}|\overline{\mathscr{E}}_{\text{target}},\mathscr{E}_{\text{SoT}_{i,j}}\} = P\{\mathscr{E}_{\text{det}}|\overline{\mathscr{E}}_{\text{target}},\mathscr{E}_{\text{SoT}_{i,j}}\} = \begin{cases} \frac{4r_s}{3cv} & \text{if } 2r_s/v < c \\ \frac{\frac{4r_s}{3} - \frac{(cv+4r_s)\sqrt{16r_s^2 - 8cr_s v}}{12r_s}}{cv} + \frac{\sqrt{16r_s^2 - 8r_s cv}}{4r_s} & \text{else.} \end{cases} \tag{C.6}$$

Since, at this point, $P\left\{\mathscr{E}_{\text{det}}^{(i)}|\mathscr{E}_{\text{SoT}_{i,j}}\right\} = P\left\{\mathscr{E}_{\text{det}}^{(j)}|\mathscr{E}_{\text{SoT}_{i,j}}\right\}$, we can rewrite expression (C.1) as:

$$P\left\{\mathscr{E}_{\text{det}}^{(i,j)}|\mathscr{E}_{\text{SoT}_{i,j}}\right\} = 2P\left\{\mathscr{E}_{\text{det}}^{(i)}|\mathscr{E}_{\text{SoT}_{i,j}}\right\} - \left(P\left\{\mathscr{E}_{\text{det}}^{(i)}|\mathscr{E}_{\text{SoT}_{i,j}}\right\}\right)^2. \tag{C.7}$$

[8] According to Fig. C.18, we should compute $L_i^{(i,j)}$ and $L_j^{(i,j)}$ by considering the angles $\Theta_i^{(i,j)}$ and $\Theta_j^{(i,j)}$, respectively. However, since the expression is the same in for both node i and node j and for ease of notation simplification, we refer to L and Θ, considering a generic pair of sensors.

[9] As for expression (C.2), for the sake of notational simplification, we refer to $\theta_{\lim}$, instead of $\theta_{\lim_1}$ and $\theta_{\lim_2}$.

At this point, the probability $P\left\{\mathscr{E}_{\text{det}}^{(i)}|\mathscr{E}_{\text{SoT}_{i,j}}\right\}$ can be analyzed following the derivation of $P\{\mathscr{E}_{\text{det}}|\mathscr{E}_{\text{SoT}}\}$ in Appendix B, but for replacing the joint pdf (B.3) with the joint pdf (C.5).

Appendix D. Derivation of $\overline{D}_{\text{det}}$ in the presence of duty cycling

Consider the trajectory identified by (x,θ). Assuming that there are $n(x,\theta)$ sensor nodes on the trajectory *and* that the target is detected along this trajectory, the probability that the target is detected by the hth sensor node is equal to the probability that the previous $(h-1)$ sensor nodes do not detect the target and the hth does—in other words, the hth sensor node along the trajectory is the first one to detect the target. In other words, denoting as $\mathscr{E}_{\text{det}}^{(x,\theta)}$ the event that the target is detected along the trajectory (x,θ), one can write:

$$\begin{aligned}&P\left\{\text{The target is detected by the } h\text{th node}|\mathscr{E}_{\text{det}}^{(x,\theta)}\right\}\\ &=\frac{P\left\{\mathscr{E}_{\text{det}}^{(h)}|x,\theta,\overline{\mathscr{E}}_{\text{det}}^{(h-1)},\ldots,\overline{\mathscr{E}}_{\text{det}}^{(1)}\right\}\prod_{k=1}^{h-1}P\left\{\overline{\mathscr{E}}_{\text{det}}^{(k)}|x,\theta,\overline{\mathscr{E}}_{\text{det}}^{(k-1)},\ldots,\overline{\mathscr{E}}_{\text{det}}^{(1)}\right\}}{\sum_{z=1}^{n(x,\theta)}P\left\{\mathscr{E}_{\text{det}}^{(z)}|x,\theta,\overline{\mathscr{E}}_{\text{det}}^{(z-1)},\ldots,\overline{\mathscr{E}}_{\text{det}}^{(1)}\right\}\prod_{v=1}^{z-1}P\left\{\overline{\mathscr{E}}_{\text{det}}^{(v)}|x,\theta,\overline{\mathscr{E}}_{\text{det}}^{(v-1)},\ldots,\overline{\mathscr{E}}_{\text{det}}^{(1)}\right\}}\quad h\\ &=1,\ldots,n(x,\theta).\end{aligned} \tag{D.1}$$

Note that the normalization comes from the fact that we are assuming that the target is detected. In fact, the probability that the target is not detected is equal to

$$1-\sum_{z=1}^{n(x,\theta)}P\left\{\mathscr{E}_{\text{det}}^{(z)}|x,\theta,\overline{\mathscr{E}}_{\text{det}}^{(z-1)},\ldots,\overline{\mathscr{E}}_{\text{det}}^{(1)}\right\}\prod_{v=1}^{z-1}P\left\{\overline{\mathscr{E}}_{\text{det}}^{(v)}|x,\theta,\overline{\mathscr{E}}_{\text{det}}^{(v-1)},\ldots,\overline{\mathscr{E}}_{\text{det}}^{(1)}\right\}$$

but we do not account for this possibility, as it is meaningless for the evaluation of the delay. Note, therefore, that looking only at the delay before detection might be misleading, as there may be a small delay because only a few targets are detected. In other words, the information on the delay before detection needs to be correlated with the probability of missed detection: this will be done in Section 4.3.

Denoting the delay associated with the detection from the hth node as $D_h = D_h(x,\theta)$, it follows that the average delay $\overline{D}(x,\theta)$ can be computed as follows:

$$\overline{D}(x,\theta)=\sum_{h=1}^{n(x,\theta)}D_hP\left\{\text{The target is detected by the } h\text{th node }|\mathscr{E}_{\text{det}}^{(x,\theta)}\right\}, \tag{D.2}$$

where *P*{The target is detected by the hth node} is is given by (D.1). The delay expression at the right-hand side of (D.2) is given by the sum of the detection delays associated with the sensors on the trajectory identified by (x,θ). In particular: the delay along this trajectory will be that of the second sensor *provided that* the first sensor misses the target; the delay will be that of the third sensor *provided that* the first and second sensors misses the target; and so on. Owing to the independence between the sensors, it follows that

$$P\left\{\mathscr{E}_{\text{det}}^{(h)}|x,\theta,\overline{\mathscr{E}}_{\text{det}}^{(h-1)},\ldots,\overline{\mathscr{E}}_{\text{det}}^{(1)}\right\}=P\left\{\mathscr{E}_{\text{det}}^{(h)}|x,\theta\right\},$$

where $P\left\{\mathscr{E}_{\text{det}}^{(h)}|x,\theta\right\}$ is derived in Appendix E.

Therefore, the average delay (D.2) along the (x,θ) trajectory can be written as

$$\overline{D}(x,\theta)=\sum_{h=1}^{n(x,\theta)}D_h\frac{P\left\{\mathscr{E}_{\text{det}}^{(h)}|x,\theta\right\}\prod_{k=1}^{h-1}P\left\{\overline{\mathscr{E}}_{\text{det}}^{(k)}|x,\theta\right\}}{\sum_{z=1}^{n(x,\theta)}P\left\{\mathscr{E}_{\text{det}}^{(z)}|x,\theta\right\}\prod_{v=1}^{z-1}P\left\{\overline{\mathscr{E}}_{\text{det}}^{(v)}|x,\theta\right\}}$$

which corresponds to (10).

Appendix E. Derivation of $P\left\{\mathcal{E}_{\text{det}}^{(k)}|x,\theta\right\}$

For the sake of notational conciseness, we simply denote $P\left\{\mathscr{E}_{\text{det}}^{(k)}|x,\theta\right\}$ as $P\{\mathscr{E}_{\text{det}}|x,\theta\}$, assuming implicitly that we are considering the kth sensor node whose sensed area is intersected by the target trajectory identified by (x,θ).

Let $\mathscr{E}_{\text{target}}$ be the event {The sensor is on at the instant at which the target enters the sensed area}. Applying the total probability theorem [32], $P\{\mathscr{E}_{\text{det}}|x,\theta\}$ can then be expressed as

$$\begin{aligned}P\{\mathscr{E}_{\text{det}}|x,\theta\}&=P\{\mathscr{E}_{\text{det}}|\mathscr{E}_{\text{target}},x,\theta\}P\{\mathscr{E}_{\text{target}}|x,\theta\}\\&\quad+P\{\mathscr{E}_{\text{det}}|\overline{\mathscr{E}}_{\text{target}},x,\theta\}P\{\overline{\mathscr{E}}_{\text{target}}|x,\theta\}\end{aligned} \tag{E.1}$$

where $P\{\mathscr{E}_{\text{det}}|\mathscr{E}_{\text{target}},x,\theta\}=1$. Since $\mathscr{E}_{\text{target}}$ is independent of the realizations of the entrance point and entrance angle of the target—in fact, the activity cycle of a sensor does not depend on the target trajectory—one can write

$$P\{\mathscr{E}_{\text{target}}|x,\theta\}=P\{\mathscr{E}_{\text{target}}\}=\int_0^{\beta_{\text{sens}}t_{\text{sens}}}\frac{1}{t_{\text{sens}}}\mathrm{d}t=\beta_{\text{sens}}$$

and, therefore, $P\{\overline{\mathscr{E}}_{\text{target}}|x,\theta\}=1-\beta_{\text{sens}}$. We are now going to evaluate the last unknown probability at the right-hand side of Eq. (E.1), i.e., $P\{\mathscr{E}_{\text{det}}|\overline{\mathscr{E}}_{\text{target}},x,\theta\}$. According to the conditioning on $\overline{\mathscr{E}}_{\text{target}}$ and x,θ, the target arrival time, denoted as T_{a}, is a random variable uniformly distributed over an interval of length $(1-\beta_{\text{sens}})t_{\text{sens}}$. To have successful detection, the target must remain in the sensed area until the sensor turns on its sensing device in the following active period. Since the target arrives with a finite speed v, the crossing time of the sensed area is $t_{\text{cross}} = l(x,\theta)/v$, where $l(x,\theta)$ is the length of the intersection between the target trajectory (identified by (x,θ)) and the area sensed by a sensor.[10]

In this case as well, depending on the given trajectory, one must distinguish between two cases: (i) $t_{\text{cross}} > (1-\beta_{\text{sens}})t_{\text{sens}}$ and (ii) $t_{\text{cross}} < (1-\beta_{\text{sens}})t_{\text{sens}}$. In the former case, the target will be detected, since it remains in the sensed area for a time interval longer than the sleeping interval, i.e., $P\{\mathscr{E}_{\text{det}}|\overline{\mathscr{E}}_{\text{target}},x,\theta\}=1$. Therefore, in this case $P\{\mathscr{E}_{\text{det}}|x,\theta\}=1$. In the latter case, instead, the target will be detected if it enters the sensed area in the last part of the sleeping interval, so that it will be detected in the following active period. The evaluation of $P\{\mathscr{E}_{\text{det}}|\overline{\mathscr{E}}_{\text{target}},x,\theta\}$ in (E.1) can be carried out as follows. Letting $\mathscr{E}_{\text{nd}}$ be the event {The target enters the sensed area while the sensing

[10] Note that $l(x,\theta)$ depends on the intersection of the target trajectory with the sensed area of the kth sensor. In other words, for different values of k, $l(x,\theta)$ is likely to change.

interface is off and it will be detected in the following duty cycle}, $P\{\mathscr{E}_{\mathrm{det}}|\overline{\mathscr{E}}_{\mathrm{target}},x,\theta\}$ can be then expressed as

$$P\{\mathscr{E}_{\mathrm{det}}|\overline{\mathscr{E}}_{\mathrm{target}},x,\theta\} = P\{\mathscr{E}_{\mathrm{det}}|\overline{\mathscr{E}}_{\mathrm{target}},x,\theta,\mathscr{E}_{\mathrm{nd}}\}P\{\mathscr{E}_{\mathrm{nd}}|\overline{\mathscr{E}}_{\mathrm{target}},x,\theta\} + P\{\mathscr{E}_{\mathrm{det}}|\overline{\mathscr{E}}_{\mathrm{target}},x,\theta,\overline{\mathscr{E}}_{\mathrm{nd}}\}P\{\overline{\mathscr{E}}_{\mathrm{nd}}|\overline{\mathscr{E}}_{\mathrm{target}},x,\theta\}, \tag{E.2}$$

where $P\{\mathscr{E}_{\mathrm{det}}|\overline{\mathscr{E}}_{\mathrm{target}},x,\theta,\overline{\mathscr{E}}_{\mathrm{nd}}\} = 0$ and $P\{\mathscr{E}_{\mathrm{det}}|\overline{\mathscr{E}}_{\mathrm{target}},x,\theta,\mathscr{E}_{\mathrm{nd}}\} = 1$. The last term at the right-hand side of (E.2) that remains to be evaluated is $P\{\mathscr{E}_{\mathrm{nd}}|\overline{\mathscr{E}}_{\mathrm{target}},x,\theta\}$. This term represents the probability that the target enters the sensed area when the sensing interface is off and it does not exit before the sensing interface switches on in the next cycle. This can be expressed as $P\{T_{\mathrm{a}} + l/v > (1-\beta_{\mathrm{sens}})t_{\mathrm{sens}} - T_{\mathrm{a}}\}$. Therefore, one can write

$$P\{\mathscr{E}_{\mathrm{nd}}|\overline{\mathscr{E}}_{\mathrm{target}},x,\theta\} = \int_{\frac{(1-\beta_{\mathrm{sens}})t_{\mathrm{sens}}-l/v}{2}}^{(1-\beta_{\mathrm{sens}})t_{\mathrm{sens}}} \frac{1}{(1-\beta_{\mathrm{sens}})t_{\mathrm{sens}}}\mathrm{d}t_{\mathrm{a}} = \frac{1}{2} + \frac{l/v}{2(1-\beta_{\mathrm{sens}})t_{\mathrm{sens}}}. \tag{E.3}$$

Using (E.2) in (E.3), $P\{\mathscr{E}_{\mathrm{det}}|\overline{\mathscr{E}}_{\mathrm{target}},x,\theta\}$ can be expressed as

$$P\{\mathscr{E}_{\mathrm{det}}|\overline{\mathscr{E}}_{\mathrm{target}},x,\theta\} = \frac{1-\beta_{\mathrm{sens}}}{2} + \frac{l}{2vt_{\mathrm{sens}}}. \tag{E.4}$$

Finally, combining (E.2) and (E.3) in (E.1), one obtains

$$P\{\mathscr{E}_{\mathrm{det}}|x,\theta\} = \begin{cases} \frac{1+\beta_{\mathrm{sens}}}{2} + \frac{l}{2vt_{\mathrm{sens}}} & \text{if } t_{\mathrm{cross}} < (1-\beta_{\mathrm{sens}})t_{\mathrm{sens}} \\ 1 & \text{otherwise}. \end{cases}$$

References

[1] M. Buettner, G. Yee, E. Anderson, R. Han, X-mac: a short preamble mac protocol for duty-cycled wireless sensor networks, in: Proc. ACM SenSys, 2006.

[2] G. Lu, B. Krishnamachari, C.S. Raghavendra, An adaptive energy-efficient and low-latency mac for data gathering in wireless sensor networks, in: Proc. Parallel and Distributed Processing Symposium.

[3] I. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, A survey on sensor networks, IEEE Communications Magazine 40 (8) (2002) 102–114.

[4] K. Bobier, J. Wells, M. Dapper, Enhanced unattended ground sensor system communications, in: Proc. MILCOM, 2004.

[5] 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, Vigilnet: an integrated sensor network system for energy-efficient surveillance, ACM Transactions on Sensor Networks 2 (1) (2006) 1–38.

[6] P. Medagliani, J. Leguay, V. Gay, M. Lopez-Ramos, G. Ferrari, Engineering energy-efficient target detection applications in wireless sensor networks, in: Proc. IEEE PerCom, 2010.

[7] P. Medagliani, J. Leguay, G. Ferrari, V. Gay, M. Lopez-Ramos, Accurate performance bounds for target detection in wsns with deterministic node placement, in: Proc. IEEE ISWPC, 2010.

[8] N. Zhao, L. Sun, Research on cross-layer frameworks design in wireless sensor networks, in: Proc. IEEE ICWMC, 2007.

[9] T. Melodia, M.C. Vuran, D. Pompili, The state of the art in cross-layer design for wireless sensor networks, in: Proceedings of EuroNGI Workshops on Wireless and Mobility, Springer Lecture Notes on Computer Science, LNCS 388, 2005.

[10] Q. Cao, T. Yan, J.A. Stankovic, T. Abdelzaher, Analysis of target detection performance for wireless sensor networks, in: Proc. DCOSS, 2005.

[11] V.Prasad, T. Yan, P. Jayach, Z. Li, S.H. Son, J.A. Stankovic, J. Hansson, T. Abdelzaher, Andes: an analysis-based design tool for wireless sensor networks, in: Proc. RTSS, 2007.

[12] Y. Ting, Analysis Approaches for Predicting Performance of Wireless Sensor Networks, Ph.D. Thesis, Charlottesville, VA, USA, Adviser – John A. Stankovic, 2006.

[13] H. Zhang, J. Hou, Is deterministic deployment worse than random deployment for wireless sensor networks? in: Proc. INFOCOM, 2006.

[14] H. Zhang, J. Hou, Maintaining sensing coverage and connectivity in large sensor networks, Journal Ad Hoc & Sensor Wireless Networks 1 (1–2) (2005).

[15] M. Younis, K. Akkaya, Strategies and techniques for node placement in wireless sensor networks: a survey, Journal of Ad Hoc Networks 6 (4) (2008) 621–655.

[16] xBow MICAz: Wireless Measurement System. <http://www.xbow.com/Products/Product_pdf_files/Wireless_pdf/MICAz_Datasheet.pdf>.

[17] J. Jeong, Y. Gu, T. He, D. Du, VISA: virtual scanning algorithm for dynamic protection of road networks, in: Proc. INFOCOM, 2009.

[18] E. Yanmaz, H. Guclu, Stationary and mobile target detection using mobile wireless sensor networks, CoRR abs/1006.4326.

[19] Y. Cao, B. Yang, Intruding target detection using wireless sensor network, in: Proc. ICEMI, 2009.

[20] L. Lazos, R. Poovendran, J.A. Ritcey, Analytic evaluation of target detection in heterogeneous wireless sensor networks, ACM Transactions on Sensor Networks 5 (2) (2009) 1–38.

[21] W. Feller, An Introduction to Probability Theory and its Applications, New York, NJ, USA.

[22] K. Klues, G. Hackmann, O. Chipara, C. Lu, A component-based architecture for power-efficient media access control in wireless sensor networks, in: Proc. ACM SenSys, 2007.

[23] J. Polastre, J. Hill, D. Culler, Versatile low power media access for wireless sensor networks, in: Proc. SenSys, 2004.

[24] W. Ye, F. Silva, J. Heidemann, Ultra-low duty cycle mac with scheduled channel polling, in: Proc. ACM SenSys, 2006.

[25] S. Kulkarni, M. Arumugam, TDMA service for sensor networks, in: Proc. ICDCS04, ADSN workshop, 2004.

[26] W. Ye, F. Silva, J. Heidemann, Ultra-low duty cycle mac with scheduled channel polling, in: Proc. ACM SenSys, 2006.

[27] B.L. Titzer, D.K. Lee, J. Palsberg, Avrora: scalable sensor network simulation, in: Proc. IPSN, 2005.

[28] R.P. Alberola, D. Pesch, Avroraz: extending Avrora with an IEEE 802.15.4 compliant radio chip model, in: Proc. ACM PM2HW2N, 2008.

[29] D. Bertsekas, R. Gallager, Data Networks, second ed., Upper Saddle River, NJ, USA, 1992.

[30] D. Bertsimas, J. Tsitsiklis, Introduction to Linear Optimization, Athena Scientific, Belmont, MA, USA, 1997.

[31] J. Sylvester, On a funicular solution of Buffon's "problem of the needle" in its most general form, Acta Mathematica 14 (1) (1890) 185–205.

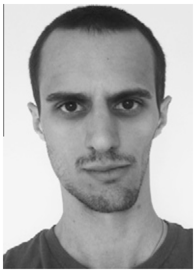
[32] A. Papoulis, Probability, Random Variables and Stochastic Processes, McGraw-Hill, New York, NJ, USA, 1991.

Paolo Medagliani was born in Cremona (CR), on July 1981. He received the "Laurea" degree in Telecommunications Engineering (3-year program) on September 2003 and the "Laurea Specialistica" degree on April 2006, respectively, from the University of Parma, Italy. From January 2007, he is a Ph.D. student in Information Technologies at the Information Engineering Department of the University of Parma, Italy. He is a member, at the Information Engineering Department of the University of Parma, Italy, of the Wireless Ad-hoc and Sensor Networks (WASN) Laboratory. His research interests are in the field of performance analysis and design of wireless sensor networks and ad hoc networks. In 2010 he got the PhD in Information Technologies at the University of Parma. Dr. Medagliani is reviewer for some international conferences (SECON 2007, WCNC 2007, GLOBECOM 2007 and 2008, WPMC 2007, EWSN 2008, ICC 2008, and PIMRC 2008) and international journals (IEEE Transactions on Wireless Communications). In addition, he is member of the technical committee of International Conference on Advances in Satellite and Space Communications (SPACOMM 2009), Colmar, France, July 19–24, 2009 and of International Workshop on Performance Methodologies and Tools for Wireless Sensor NetworksWSN-Perf (WSNPerf 2009), Pisa, Italy, October 23, 2009. Dr. Medagliani has also taken part into research projects in collaboration with a few research organizations and private companies. As for now Dr. Medagliani is working in Lepida s.p.a. in Bologna as a project manager.

Gianluigi Ferrari was born in Parma, Italy, in 1974. He received his "Laurea" and PhD degrees from the University of Parma, Italy, in 1998 and 2002, respectively. Since 2002, he has been with the University Parma, where he currently is an Associate Professor of Telecommunications. He was a visiting researcher at USC (Los Angeles, CA, USA, 2000-2001), CMU (Pittsburgh, PA, USA, 2002-2004), KMITL (Bangkok, Thailand, 2007), and ULB (Bruxelles, Belgium, 2010). Since 2006, he has been the Coordinator of the Wireless Ad-hoc and Sensor Networks (WASN) Lab in the Department of Information Engineering of the University of Parma.

As of today he has published more than 140 papers in leading international journals and conferences. He is coauthor of a few books, including *Detection Algorithms for Wireless Communications, with Applications to Wired and Storage Systems* (Wiley: 2004), *Ad Hoc Wireless Networks: A Communication-Theoretic Perspective* (Wiley: 2006), *LDPC Coded Modulations* (Springer: 2009), and *Sensor Networks with IEEE 802.15.4 Systems: Distributed Processing, MAC, and Connectivity* (Springer: 2011). He edited the *book Sensor Networks: Where Theory Meets Practice* (Springer: 2010). His research interests include digital communication systems analysis and design, wireless ad hoc and sensor networking, adaptive digital signal processing, and information theory.

Prof. Ferrari is a co-recipient of a best student paper award at IWWAN'06 and a best paper award at EMERGING'10. He acts as a frequent reviewer for many international journals and conferences. He acts also as a technical program member for many international conferences. He currently serves on the Editorial Boards of *The Open Electrical and Electronic Engineering (TOEEJ) Journal*(Bentham Publishers), the *International Journal of RF Technologies: Research and Applications* (Taylor & Francis), and the *International Journal of Future Generation Communication and Networking* (SERSC: Science & Engineering Research Support Center). He was a Guest Editor of the 2010 EURASIP JWCN Special Issue on "Dynamic Spectrum Access: From the Concept to the Implementation."



Vincent Gay got the diploma in Networking from the Ecole Nationale Supérieure des Télécommunications de Paris in 2007. He entered the Laboratory for Advanced Information Technologies of Thales Communications France in 2007. He is working on energy-efficiency strategies in Wireless Sensor Networks in ITEA2 GEODES project, and on the integration of WSN into objecttoobject systems in ITEA2 USENET project. He has also been involved in studies on networking protocols for mobility and multi-homing (e.g., Mobile IPv6, HIP) in the context of FP6/IST ANEMONE and MULTINET, and French-RNRT REMORA.

Jérémie Leguay is an advanced studies engineer at Thales Communications, in Colombes, France. He received his Engineering Degree (2003) from EFREI (École Française d'Electronique et d'Informatique) and a Master of Science (2004) in Computer Science from Linköping University in Sweden. From 2004 to 2007 he has been a Ph.D. candidate at the Computer Science laboratory (LIP6) of Pierre & Marie Curie University and at Thales Communications where he conducted research in ad hoc and delaytolerant networking. His current research interests focus on networking solutions for mobile and constrained environments.