Computer Networks 79 (2015) 216–235

Contents lists available at ScienceDirect

Computer Networks

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

Effective and efficient neighbor detection for proximity-based mobile applications

Behnaz Bostanipour *, Benoît Garbinato

Distributed Object Programming Laboratory, Internef Building, University of Lausanne, CH-1015 Lausanne, Switzerland

ARTICLE INFO

Article history:
Received 15 April 2014
Received in revised form 13 November 2014
Accepted 17 December 2014
Available online 6 January 2015

Keywords:
MANET
Neighbor detection
IEEE 802.11
Smartphone

ABSTRACT

We consider the problem of maximizing both *effectiveness* and *efficiency* of the detection of a device by another device in a mobile ad hoc network, given a maximum amount of time that they remain in the proximity of each other. Effectiveness refers to the degree to which the detection is successful, while efficiency refers to the degree to which the detection is energy saving. Our motivation lies in the emergence of a new trend of mobile applications known as proximity-based mobile applications which enable a user to communicate with other users in some defined range and for a certain amount of time. The highly dynamic nature of these applications makes neighbor detection time-constrained, i.e., even if a device remains in proximity for a limited amount of time, it should be detected with a high probability as a neighbor. In addition, the limited battery life of mobile devices requires the neighbor-detection to be performed by consuming as little energy as possible. To address this problem, we perform a realistic simulation-based study in mobile ad hoc networks and we consider three typical urban environments where proximity-based mobile applications are used, namely *indoor with hard partitions*, *indoor with soft partitions* and *outdoor urban areas*. In our study, a node periodically broadcasts a message in order to be detected as a neighbor. Thus, we study the effect of parameters that we believe could influence effectiveness and efficiency, i.e., *the transmission power* and *the time interval between two consecutive broadcasts*. Our results show that regardless of the environment, effectiveness and efficiency are in conflict with each other. Thus, we propose a metric that can be used to make good tradeoffs between effectiveness and efficiency.

1. Introduction

With the increasing use of mobile devices and particularly smartphones, we face the emergence of a new blend of distributed applications known as *Proximity-Based Mobile* (*PBM*) applications [10–12]. These applications enable a user to interact with others in a defined range and for a given time duration e.g., for social networking (WhosHere [53], LoKast [31]), gaming (Bluetooth gaming apps [8]) and driving (Waze [52]).

Discovering who is nearby is a basic requirement of various PBM applications. In a simple usage scenario of social networking applications such as WhosHere [53] or LoKast [31], a user can discover other users in a defined range, view their profiles and chat with a user or a group of users with her phone. Usually, the highly dynamic nature of these applications (which is basically due to the mobility of devices) makes neighbor detection time-constrained, i.e., even if a device remains in proximity for a limited amount of time, it should be detected with a high probability as a neighbor. In addition, the limited battery life of

* Corresponding author. Tel.: +41 21692 34 10.
E-mail addresses: behnaz.bostanipour@unil.ch (B. Bostanipour), benoit.garbinato@unil.ch (B. Garbinato).

http://dx.doi.org/10.1016/j.comnet.2014.12.013
1389-1286/

mobile devices requires the neighbor-detection to be performed by consuming as little energy as possible.

In this paper, we consider the following problem: *how can a device be detected by another device with both maximum effectiveness and maximum efficiency, given a maximum amount of time that they remain in proximity of each other? If not, how can an effectiveness-efficiency tradeoff be made?* Effectiveness refers to the degree to which the detection is successful and is measured by the detection probability, while efficiency refers to the degree to which the detection is energy saving and is measured by the inverse of energy consumption per device. To address this problem, we evaluate effectiveness and efficiency in a single-hop mobile ad hoc network (MANET). The evaluations are performed under realistic assumptions and based partly on simulations using the ns-2 [37] network simulator.[1]

There are two main reasons behind our choice of a MANET as the underlying network architecture. Firstly, MANETs seem to be the most natural existing technology to enable PBM applications. In fact, similarly to PBM applications, in a MANET two nodes can communicate if they are within a certain distance of each other (to have radio connectivity) for a certain amount of time. Secondly, mobile devices are increasingly equipped with ad hoc communications capabilities (e.g., WiFi in ad hoc mode or Bluetooth) which increases the chance of MANETs to be one of the future mainstream technologies for PBM applications.

Since the quality of radio signals (and consequently the detection probability) is affected by the environment attenuation, for our study we consider three typical urban environments where PBM applications are used, i.e., *indoor with hard partitions* (corresponding to offices with thick walls), *indoor with soft partitions* (corresponding to exhibitions with temporary partitions) and *outdoor urban areas* (corresponding to a music festival in downtown). To simulate these environments, we use a radio propagation model known for modeling the obstructed urban environments called *Log-Normal Shadowing* (LNS).

In our study, a node periodically broadcasts a *hello* message during a fixed time interval in order to be detected as a neighbor. We assume that the nodes use the *IEEE 802.11a* standard for the physical and MAC layer. Thus, we study the impact of two key parameters that influence effectiveness and efficiency, i.e., *the transmission power* and *the time interval between two consecutive broadcasts*. In performing the evaluations, we are particularly interested to answer the following questions:

- *In each environment, when does a change in the value of any of the above mentioned parameters increase effectiveness and efficiency, or on the contrary, when does it deteriorate them?*
- *In each environment, is there a unique combination of these parameters that could maximize both effectiveness and efficiency? If not, how could a tradeoff between effectiveness and efficiency be made?*

[1] The version 2.35 (the latest version), released on November 4, 2011.

1.1. Contributions and roadmap

This paper is, to the best of our knowledge, the first study on the impact of transmission power and broadcast interval on effectiveness and efficiency of neighbor detection for MANETs in urban environments. It provides a detailed simulation study and defines the metrics that can be used to interpret the results. In order for our results to be close to reality, the study is performed under realistic assumptions. For one thing, we use 802.11a technology for communication between nodes and we assume a probabilistic radio propagation model for urban environments. Furthermore, we calculate the energy consumption using the specification of typical smartphones.

The remainder of the paper is as follows. In Section 2, we describe our system model. In particular, we define the *neighbor detection algorithm*, which takes transmission power and broadcast interval (this pair constitutes a *strategy*) as input. In Section 3, we formulate the problem studied in this paper. It basically consists of finding the most effective and the most efficient strategy in each environment. If these strategies are not equal in an environment, we intend to find a strategy that makes a reasonable tradeoff between effectiveness and efficiency. We also define the set of strategies for which the effectiveness and efficiency are evaluated. In Section 4, we evaluate the effectiveness for the set of predefined strategies. We also discuss the impact of changing transmission power and broadcast interval on effectiveness. Finally, we identify the most effective strategy in each environment. In Section 5, we evaluate the efficiency for the set of predefined strategies. We show that efficiency is independent of the environment and we discuss the impact of changing transmission power and broadcast interval on efficiency. Finally, we identify the most efficient strategy. In Section 6, we compare the results of Sections 4 and 5. We observe that we cannot find a strategy that maximizes both effectiveness and efficiency in any environment. The reason is that, regardless of environment, effectiveness and efficiency are in conflict with each other. We then propose an approach to make a tradeoff between effectiveness and efficiency. Using this approach, we find the tradeoff strategy in each environment and we show that it has a relatively good effectiveness and efficiency compared to other strategies. Finally, we discuss related work in Section 7 before concluding in Section 8 with a perspective on future work.

2. System model

In this section, we present the system model, and whenever necessary, we describe the reasons behind our modeling choices.

2.1. Processes

We consider a mobile ad-hoc network (MANET) consisting of a finite set of n processes $P = \{p_1, \ldots, p_n\}$. We use the terms *process* and *node* interchangeably. Processes are in a two-dimensional plane. Each process has a unique identifier and is aware of its own geographic location at any time. Processes can experience *crash* failures. A crash faulty process stops prematurely. Prior to stopping, it behaves correctly. Since we do not consider *Byzantine* behaviors, information security and privacy issues are beyond the scope of this paper.

2.2. Time

We assume the existence of a discrete global clock, i.e., the clock's tick range is the set of non-negative integers. Every process has a local clock which has the same clock's tick range as the global clock and runs at the same rate as the global clock, but its time value has some offset from the global time.

2.3. Communication

We consider a single-hop network i.e., without any message routing mechanism. Processes communicate by broadcasting messages using the *IEEE 802.11a* MAC and physical layers [2,28]. The current WiFi technology used in mobile devices is based on three IEEE standards, i.e., *802.11a*, *802.11b*, *802.11g*. There are two main reasons for our choice of 802.11a over the other standards: (1) 802.11b/g operate in the 2.4 GHz frequency band which is heavily used not only by WiFi devices but also by other devices such as microwave ovens and DECT phones whereas, 802.11a operates in the relatively unused 5 GHz frequency band. Thus, using 802.11a results in less interference and better throughput. This makes 802.11a an appealing technology for ad hoc communication in urban areas where PBM applications are mostly used; (2) the most recent IEEE 802.11 standard, i.e., *802.11ac* also operates in 5 GHz frequency band [3] and uses some similar modulation schemes and coding rates for broadcast as 802.11a. Thus, using 802.11a allows us to have the results which are close to those that could be obtained with the new standard.

Finally, we assume that each process has a buffer (a queue) that stores messages after their generation and before their broadcast. The size of the queue and the messages are such that the queue never remains full long enough to cause it to drop a message.

2.4. Environment

We consider three typical urban environments where PBM applications are used, namely *indoor with hard partitions* (corresponding to offices with thick walls), *indoor with soft partitions* (corresponding to exhibitions with temporary partitions) and *outdoor urban areas* (corresponding to a music festival in downtown). In our study, we use a probabilistic model called the *Log-Normal Shadowing* (*LNS*) for the radio propagation in an urban environment [40]. LNS uses a log-normal random variable to describe the variations of the received power and has two parameters, i.e., *the path loss exponent* (β) and *the shadowing deviation* (σ) to characterize each environment. The path loss exponent (β) captures the average signal attenuation due to effects such as absorption, refraction, diffraction, reflection, etc. The shadowing deviation (σ) captures the radio irregularity. If ($\sigma = 0$), the radio propagation range is a perfect circle, but as σ grows, its shape changes from a circle to a more random and irregular shape which reflects what happens in reality, i.e., in the presence of not perfectly isotropic antennas and the obstacles that cut the transmission range [36]. For our evaluations, we consider a distinct pair of (β, σ) values for each environment (see Table 1). These pairs are chosen based on the measurements in the literature [40].

Table 1
Values of LNS parameters for each environment.

Environment	β	σ (dB)
Indoor-hard partitions	5.5	7
Indoor-soft partitions	5	9.6
Outdoor-urban	4	5.5

2.5. Neighbor detection algorithm

Each process p_i executes the *neighbor detection algorithm*. The algorithm has two input parameters: the time duration Δ_{period} and the transmission power pow_{tx}. There is also a constant R which defines the detection range. The algorithm divides time into rounds of Δ_{period}. At the beginning of each round, p_i broadcasts a *hello* message containing the tuple $(i, roundNo, loc)$ where *roundNo* is the number of the current round and *loc* is the location of p_i at time when *hello* is sent.

When a process p_j receives a *hello* message sent by p_i, it verifies if its distance to p_i is less than or equal to R. If it is the case, p_i is detected as a neighbor at its round *roundNo* by p_j. This means that if p_i is in the neighborhood of p_j since its first round of broadcasting the *hello* message, we can say that p_i is detected after being in the neighborhood of p_j for time duration of $roundNo \times \Delta_{period}$. Note that here we ignore the elapsed time between the sending and the reception of the *hello* message, which obviates the use of a time synchronization algorithm. Also, since we consider a single-hop network, p_j can only detect p_i as a neighbor if R is smaller than or equal to the actual transmission range of p_i.

In what follows, for simplicity's sake, we designate by the term *strategy* an ordered pair $(pow_{tx}, \Delta_{period})$ which can be considered as a possible input of the neighbor detection algorithm.

3. Problem statement

We characterize neighbor detection by two main aspects:

- *Effectiveness* is defined as the degree to which the neighbor detection is successful and is measured by the detection probability. Thereby, maximizing effectiveness boils down to maximizing the detection probability.
- *Efficiency* is defined as the degree to which the detection is energy saving and is measured by the inverse of energy consumption per process. Thereby, maximizing efficiency boils down to minimizing the energy consumption per process.

Thus, the problem we consider in this paper can be specified as follows. Let $\Delta_{neighborhood}$ be the maximum amount of time that a node p_i remains continuously within the detection range R of a node p_j, then, for each environment, our goal is to find:

- *the most effective strategy*, i.e., the strategy that maximizes the effectiveness of detection of p_i by p_j;
- *the most efficient strategy*, i.e., the strategy that maximizes the efficiency of detection during $\Delta_{neighborhood}$;
- *the tradeoff strategy*, i.e., the strategy that makes a tradeoff between effectiveness and efficiency in the case that the most effective strategy is not the same as the most efficient strategy.

We address the problem by evaluating the effectiveness and the efficiency for a set S of predefined strategies and $\Delta_{neighborhood} = 4$ s.

Since the majority of current PBM applications run on smartphones, for defining the strategies in S and later in our evaluations, we use the characteristics of current smartphones. For instance, regarding power consumption, we use the specifications of Broadcom's wireless network interface cards [13]. In fact, in a mobile device, a *wireless network interface card* (denoted by *WNIC*) is the component that implements the MAC and the physical layers of the OSI model. Broadcom's WNICs are one of the most used WNICs in the current mobile devices and specially smartphones, e.g., Broadcom's BCM 4330 WNIC is used in both Samsung Galaxy S II and iPhone 4S [4].

Thus, let strategy $s = (pow_{tx}, \Delta_{period})$ be an element of S. Then, pow_{tx} can take a value of 15 dBm, 19 dBm or 25 dBm. The first two values are based on specifications of Broadcom's BCM 4329 and BCM 4330 WNICs, whereas the last value presents the possible performance gain of more powerful radio transmitters [49]. Also, Δ_{period} can take a value of 1 s, 1/2 s, 1/4 s or 1/12 s. These values have been selected after our preliminary tests which show that they can present our results in a useful manner. Considering all the combinations of the above mentioned values of pow_{tx} and Δ_{period}, the set S contains 12 strategies.

The reason behind the choice of 4 s for $\Delta_{neighborhood}$ is that current PBM applications usually guarantee the detection of a person even if she remains in neighborhood for a very limited amount of time. Therfore, we choose $\Delta_{neighborhood}$ to be reasonably short.

We also assume that each *hello* message has a size of 500 bytes. This value is chosen by considering the possibility that each message can be digitally signed and accompanied by a certificate to authenticate the sender (i.e., using a similar mechanism for message authentication as the one used for safety messages in vehicular ad hoc networks [41]). As stated earlier, in this paper we do not study the security and privacy issues. However, we choose the messages to be large enough so that our results remain valid for more general cases.

4. Evaluation of effectiveness

Effectiveness is measured by the neighbor detection probability. Thus, in this section we first describe our approach to calculate the detection probability of each strategy, which is based on simulations. Then, we present our simulation setup and the results. In particular, while presenting the results, we discuss how a change in pow_{tx} or Δ_{period} can affect the detection probability in each environment. We also define two packet dropping metrics that we use to interpret the results. Finally, we compare the effectiveness of the strategies and present the most effective strategy in each environment.

4.1. Approach to calculate the neighbor detection probability

We calculate the resulting detection probability of each strategy by performing simulations with ns-2 simulator. Each simulation takes a strategy s and an environment e as the input and produces as the output the detection probability at time t for all $t \in [0, \Delta_{neighborhood}]$.

More precisely, let $s = (pow_{tx}, \Delta_{period})$ and $e = (\beta, \sigma)$, at the beginning of a simulation we initialize the *neighbor detection algorithm* at all nodes with pow_{tx} and Δ_{period}. We also initialize the radio propagation model with β and σ. Since the value of $\Delta_{neighborhood}$ is the same while testing different strategies, instead of using nodes with movement, we consider static nodes which broadcast the *hello* message only during $\Delta_{neighborhood}$. However, since each node's local clock has some offset with respect to the global clock, nodes do not start and finish broadcasting at the same time. We only verify the detection probability at certain predefined nodes called *Reference Nodes* or *RN*s. Intuitively, the longer a node remains in the neighborhood, the more it broadcasts the *hello* message, and thus it has more chance to be detected. Therefore, the neighbor detection probability at each RN is calculated as a cumulative probability and is an increasing function of time during which a node remains in the neighborhood. More precisely, given time $t \in [0, \Delta_{neighborhood}]$, the neighbor detection probability at t for a RN is equal to the number of all nodes that are detected by RN after being in the neighborhood for time t, divided by the number of all nodes that are in the neighborhood. Finally, since we use a probabilistic radio propagation model, for each $t \in [0, \Delta_{neighborhood}]$, we take the average of the detection probability at t over all RNs. As the result, we have the values of the average of the detection probability during $[0, \Delta_{neighborhood}]$ which constitute the output of the simulation.

Note that since we use a probabilistic radio propagation model, two simulations that have the same strategy and environment as the input do not necessarily result in the same output. Thereby, to have a good estimation of the corresponding detection probability of a strategy s in an environment e, we take the average of the outputs of five simulations which have s and e as the input. We explain this in more detail in Section 4.2 after defining the simulation setup.

4.2. Simulation setup

We choose the total number of nodes and the location of RNs such that we can estimate the detection probability in the worst case, i.e., where the communication interference and collisions are at the maximum. In fact, if a strategy maximizes the neighbor detection probability in the

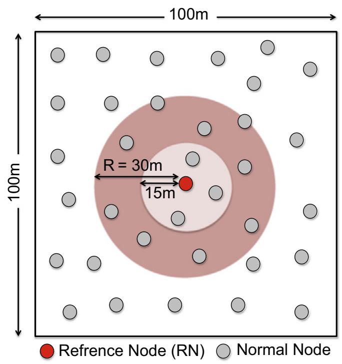

Fig. 1. Simulation map.

worst case, we can assume that it maximizes the neighbor detection probability in all cases.

Thus, we consider a square of 100 m width filled with 1000 static nodes located using a uniform random distribution.[2] RNs are the nodes located at the distance less than or equal to 5 m from the center of the square. The reason is that the nodes that are close to the center, usually experience the maximum radio interference. The total number of nodes is chosen after studying the *occupancy load factors* of urban surfaces [16]. In architecture and urbanism, the occupancy load factor of a given urban surface defines the maximum number of persons which can occupy one unit area of that surface. The occupancy load factor of an urban surface is mainly defined based on its usage (e.g., residential, office, public assembly, etc.). Although, the occupancy load factors of the urban environments that we consider in this paper are slightly different from one another, 1000 seems to be a good approximation of the maximum number of persons that usually occupy these environments.

For the radio propagation model, we use the implementation of LNS model in ns-2. In a simulation, all nodes have the same idealized transmission range[3] since they are all initialized with a given value of pow_{tx}. Moreover, even with our lowest pow_{tx} choice, the idealized transmission range is large enough so that all nodes are within the idealized transmission range of each other. Each RN has the detection range of 30 m. This value is chosen such that even using our lowest pow_{tx} choice, the detection range is less than the idealized transmission range. Fig. 1 depicts the simulation map with only one RN.

For the implementation of 802.11a in ns-2, we use the implementation performed by a team from Mercedes-Benz Research and Development North America and University of Karlsruhe [15]. This implementation includes a completely revised and enhanced architecture for physical and MAC layers to improve the drawbacks of the 802.11 default support in ns-2. In particular, this implementation for the physical layer comprises cumulative *received signal power over noise* (SINR) computation. Its MAC layer also accurately implements the CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance) mechanism.

Thus, we use the default values of this implementation for physical and MAC layers [15], however, we disable both *preamble* and *frame body* capture features.[4] For data rate, we consider 6 Mbps using *Binary Phase-Shift Keying* (*BPSK*) modulation scheme and 1/2 coding rate. In fact, more advanced schemes imply higher data rates but also require better received signal quality which reduces the number of receivable packets in the case of 802.11 broadcasts where no acknowledgment or RTS/CTS (Request to Send/Clear to Send) mechanism exist to cope with interferences and collisions.

Based on our assumptions in Section 3, the size of a *hello* packet is set to 500 bytes. If messages are generated while previously generated messages are not yet transmitted, the new messages are stored in an interface queue that is capable of storing up to 100 packets. This queue size is chosen such that regardless of the used strategy, no message is dropped by the queue.

To obtain the detection probability for a strategy s in an environment e, we perform five simulations with s and e as the input and with five different pairs of seeds. In fact, in each simulation one seed is used to initialize the random number generator of the LNS model and the other is used to initialize the random number generator responsible for the randomness of topologies. Thereby, for each simulation, we have a different topology (with different RNs) and a LNS model which is seeded differently. Then, we take the average of the five simulations' outputs. The result is considered as the detection probability for the strategy s in the environment e. Recall that the output of a simulation is the average (over all RNs in that simulation) of the neighbor detection probability at time t for all $t \in [0, 4]$ s. Therefore, the detection probability for the strategy s in the environment e is the average (over all RNs in the five simulations) of the neighbor detection probability at time t for all $t \in [0, 4]$ s.

Note that according to our preliminary tests, with five simulations we already achieve an average that accurately presents the detection probability in each setting. In fact, with five simulations the resulting standard deviation in all cases is very low (in order of 10^{-2}).

[2] In urban environments, a node's movement may depend on another node's movement (e.g., if they move in a group). A node can also move on predefined paths or sidewalks [5]. Therefore, topology changes are not always random. However, using a distribution that characterizes such behaviors, requires a particular deployment scenario, i.e., the physical properties of the terrain, the points of interest, etc. Thus, we believe that a random distribution gives a good generic basis for the evaluation.

[3] *The idealized transmission range* corresponds to the deterministic transmission range calculated for idealized deterministic channel conditions i.e., with no node movement and no obstacles between the sender and the receiver(s).

[4] *Capture* feature, which is present in some WNICs, can mitigate the effect of collisions to some degrees. Roughly speaking, when a packet collision happens, the capture feature enables the receiver to capture one of the collided packets if certain conditions are fulfilled. The existing WNICs differ in the extent of supporting capturing techniques [29,15]. There exist two variants of the capture feature (i.e., the *preamble* and the *frame body* capture) in the simulator that we use. However, according to our preliminary tests, enabling these features only increases the chance of packet reception with the same percentage across different strategies and thus, does not influence our conclusions.

4.3. Results

According to our simulation results, a node which is situated at a maximum distance of 15 m from a RN, is detected with a high probability (at least 0.8) in all environments. The reason is that at close distances (i.e., up to 15 m), in all environments, the signal strength of a received packet is usually high enough to resist interferences. Therefore, in this section we only discuss the detection of the nodes situated at a distance between 15 m to 30 m from a RN (see Fig. 1).

Fig. 2, depicts, as an example, the results for Strategy (15 dBm, 1/4 s) in different environments. As shown in the figure and already described in Section 4.1, the neighbor detection probability is an increasing function of time. We also observe that for the same strategy, the detection probability increases as we change the environment from the *indoor with hard partitions* to *indoor with soft partions* and then to *outdoor urban*. This is because the radio signals are attenuated the most in *indoor with hard partitions* and the least in *outdoor urban*.

The fact that the neighbor detection probability is calculated as a cumulative probability and is an increasing function of time enables us to only take into account its last value (i.e., the value at second 4) when comparing the effectiveness of different strategies. Thereby, for the sake of simplicity, we use henceforth the term *neighbor detection probability* while referring to the *neighbor detection probability at second 4*.

Intuitively, in a given environment increasing pow_{tx} and decreasing Δ_{period} should lead to the most effective strategy. In fact, by increasing pow_{tx}, the packets are transmitted with a more powerful signal and accordingly they better survive the interferences and environmental attenuations. Also, decreasing Δ_{period} increments the total number of sent *hellos* and thus increases the chance of reception. However, simulations show that this is not always true i.e., changing the values of pow_{tx} and Δ_{period}, will not always affect the detection probability in all environments in the same way. However, in certain cases we observe similar behaviors for some range of values, e.g., increasing pow_{tx} for a given Δ_{period} seems to increase the detection probability for the majority of cases (see Fig. 4), whereas decreasing Δ_{period} for a given pow_{tx} might result in unpredictable behaviors (see Fig. 5). To understand the reason behind these similarities and differences, we study the mechanism of packet drops by 802.11 physical layer. Based on our study, we define two metrics that can be used to interpret the results in each environment. In the following, we first present the metrics and we show, as an example, how they can be used to interpret the results in one particular environment, i.e., *indoor with hard partitions*. Then, based on the interpretation of the results of different environments using our metrics, we present our general observations on how a change in pow_{tx} or Δ_{period} can affect the detection probability.

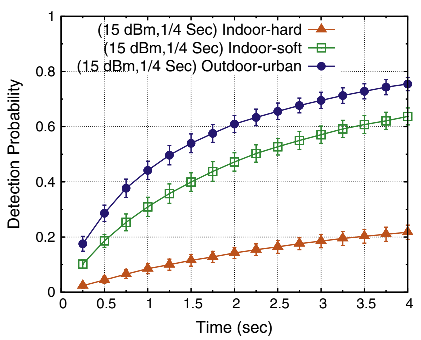

Fig. 2. Same strategy in different environments. The vertical error bars present the standard deviation for the detection probability.

4.3.1. Packet dropping metrics

Before defining our metrics, we present an overview of the packet dropping mechanism by 802.11 physical layer. When a packet arrives from the channel to the physical layer of the receiver, its *received signal power over noise* (SINR) is compared to a constant threshold called *SINR threshold*.[5] The threshold value depends on the modulation scheme and the coding rate.[6] If there is only one sender, the noise is equal to the noise floor, which is the sum of the thermal noise of the system plus some additional noise caused by losses in the receiver hardware (e.g. in the antenna cables or electronic parts) [44,15]. However, if there are other senders which send at the same time, their packets could be sensed by the receiver and increase the noise. If the SINR of a packet is less than the threshold, no reception process is triggered and the packet is dropped. A packet could also be dropped due to collisions. In this case, a packet is in the reception process, but another packet arrives. If the second packet is strong enough to corrupt the first packet by augmenting the background noise, both the first and the second packets are dropped, otherwise the second packet is dropped and the first packet reception continues. Thus, to explain our simulation results we define two following metrics.

- *Weak Packets Percentage* (*WPP*). This is the average percentage of all *weak* packets that arrive to a RN's physical layer out of all sent packets by nodes in a distance of 15 m to 30 m through a simulation. By weak packets, we mean packets which have a low power when they arrive to the physical layer such that even without any interference from other nodes, their SINR is lower than the SINR threshold.
- *Interfered or Collided Packets Percentage (ICPP)*. This is the average percentage of all interfered or collided packets out of all sent packets by nodes in a distance of 15 m to 30 m to a RN through a simulation. By

[5] The packet dropping mechanism described in this section is drawn from the *SINR-based reception model* which is adopted by many network simulators (including the simulator that we use). In this model the SINR threshold values are obtained by experimental measurements using real hardware. For more information about this model see [7,15].

[6] Different modulation schemes and coding rates can be used while transmitting the PLCP (Physical Layer Convergence Procedure) header and the data frame of a 802.11a physical layer packet. Accordingly, the SINR threshold used at the reception can be different for the PLCP header and the data frame. However, since in our simulations we use the same modulation scheme (i.e., BPSK) and coding rate (i.e., 1/2) for both the PLCP header and the data frame, in our description we only consider one SINR threshold.

interfered packets, we mean all packets which have an acceptable SINR for reception if there is no interference but their SINR is lower than the SINR threshold because of the interference of other nodes. By collided packets, we mean all packets which are dropped due to collision.

WPP and ICPP are not disjoint i.e., there are packets which are weak but have collided with the reception of another packet. WPP is a function of pow_{tx} and the environment attenuation (characterized by LNS parameters). ICPP is a function of $\Delta_{period}, pow_{tx}$ and the environment attenuation. A sent packet could also be dropped if it arrives when the receiver's physical layer is in transmission state. However, according to our preliminary evaluations, the percentage of such packets is very low, so we simply ignore them. In addition, in our preliminary experiments, we defined other metrics which are not related to the packet dropping, e.g., average backoff time at senders, however WPP and ICPP seem to interpret the results in a more clear way.

4.3.2. Metrics-based interpretation of the results

We now interpret the results for *indoor with hard partitions* environment using our defined metrics. By using this example, we show how these metrics can help us to understand the behavior of the detection probability under different strategies. Our discussion is based on the measurements depicted in Fig. 3, Fig. 4A and Fig. 5A. In particular, in Fig. 3, the values of the defined metrics for different strategies in *indoor with hard partitions* environment are presented.

- *Increasing pow_{tx} for a given Δ_{period}*. As shown in Fig. 3, as we increase pow_{tx} from 15 dBm to 19 dBm and then to 25 dBm, the value of WPP decreases, which means that the percentage of weak (non-receivable) packets that arrive to the physical layer of the receiver decreases. On the other hand, as we increase pow_{tx}, for the same Δ_{period}, the value of ICPP increases. The reason is that increasing pow_{tx} results in more powerful packets arriving to the physical layer, which can interfere or collide with other packets' reception. So, we observe that when we increase pow_{tx} considerably, i.e., from 15 dBm or 19 dBm to 25 dBm (recall that dBm is a logarithmic scale), the detection probability increases regardless of the value of Δ_{period} (see Fig. 4A). However, when we increase pow_{tx} from 15 dBm to 19 dBm, the detection probability improves differently under different values of Δ_{period}. For instance, as shown in Fig. 4A, when $\Delta_{period} = 1/12$ s, increasing pow_{tx} from 15 dBm to 19 dBm does not improve the detection probability as much as it improves under $\Delta_{period} = 1$ s. The reason is that under small values of Δ_{period}, the value of ICPP is high i.e., many packets are dropped because of collisions and interferences and therefore a small increase in pow_{tx} cannot improve the detection probability significantly.
- *Decreasing Δ_{period} for a given pow_{tx}*. As depicted in Fig. 3, the value of WPP remains the same when decreasing Δ_{period}. This is not surprising since WPP is a function of pow_{tx} and the environment attenuation and is independent from Δ_{period}. On the other hand, when decreasing Δ_{period}, the value of ICPP increases since more packets arrive per second to the physical layer of the receiver, which increases the chance of collisions and interferences. However, collisions do not have the same impact in the presence of different values of WPP. For instance, as shown in Fig. 3, when $pow_{tx} = 15$ dBm the value of WPP = 96%. In this case, even if the number of collisions increases, a large number of collided packets will be weak (non-receivable) packets. Therefore, decreasing Δ_{period} increments the reception chance of the powerful packets. As depicted in Fig. 5A, with $pow_{tx} = 15$ dBm, the detection probability at $\Delta_{period} = 1/12$ s is greater than the detection probability at $\Delta_{period} = 1$ s and almost equal to the detection probability at $\Delta_{period} = 1/4$ s. However, when the value of WPP is relatively low, collisions have a more significant impact and can decrease the detection probability e.g., as shown in Fig. 3, when $pow_{tx} = 25$ dBm, the value of WPP = 75%. In this case, the detection probability at $\Delta_{period} = 1/12$ s is even less than the detection probability at $\Delta_{period} = 1$ s (see Fig. 5A).

4.3.3. Impact of changing pow_{tx} and Δ_{period} on neighbor detection probability

After interpreting all results by using the packet dropping metrics, we reach the following general observations regarding the impact of increasing pow_{tx} and decreasing Δ_{period} on the detection probability.

- *Increasing pow_{tx} for a given Δ_{period}*. For a fixed Δ_{period}, increasing pow_{tx} considerably, i.e., from 15 dBm or 19 dBm to 25 dBm, increases the detection probability in all environments (see Fig. 4). Increasing pow_{tx} from 15 dBm to 19 dBm, improves the neighbor detection under high values of Δ_{period} (e.g., for 1 s), but under low values of Δ_{period} (e.g., for 1/12 s), it has less effect and can even lead to no improvement. For instance, as shown in Fig. 4B, in *indoor with soft partitions* environment, under $\Delta_{period} = 1$ s, increasing power from 15 dBm to 19 dBm increases the detection probability from 0.4612 to 0.6539 (i.e., about 41.78% increase), whereas under $\Delta_{period} = 1/12$ s, increasing pow_{tx} from 15 dBm to 19 dBm has almost no influence on the detection probability. This is because under low values of Δ_{period}, the number of collisions and interferences is relatively high.
Furthermore, although in all environments, increasing pow_{tx} improves the detection probability, in general, the improvement becomes less significant as we move from a more obstructed environment to a less obstructed environment. For instance, under $\Delta_{period} = 1$ s, increasing pow_{tx} from 15 dBm to 25 dBm in the *indoor with hard partitions* environment increases the detection probability by a factor of 5.66 (see Fig. 4A), whereas, in the *outdoor-urban* environment, it increases the detection probability by a factor of 1.13 (see Fig. 4C). In fact, as the environment becomes less obstructed, packets do not need to be transmitted with a very powerful signal to resist the environmental attenuation, thus, increasing pow_{tx} results in a less significant improvement of the detection probability.

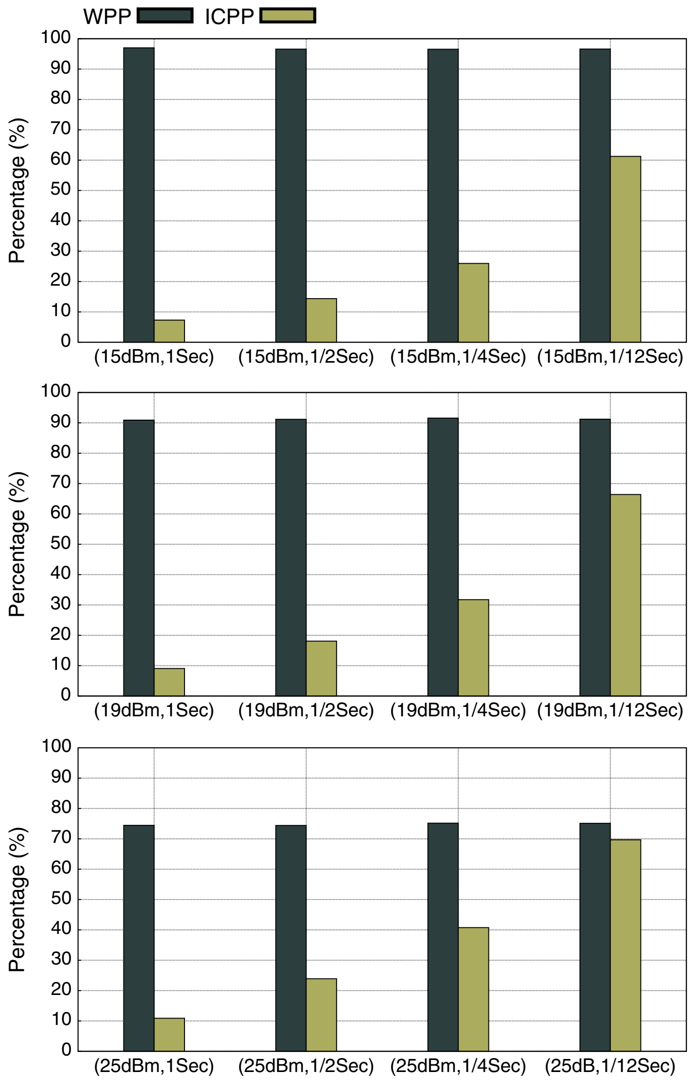

Fig. 3. Values of packet dropping metrics for different strategies in the indoor with hard partitions environment.

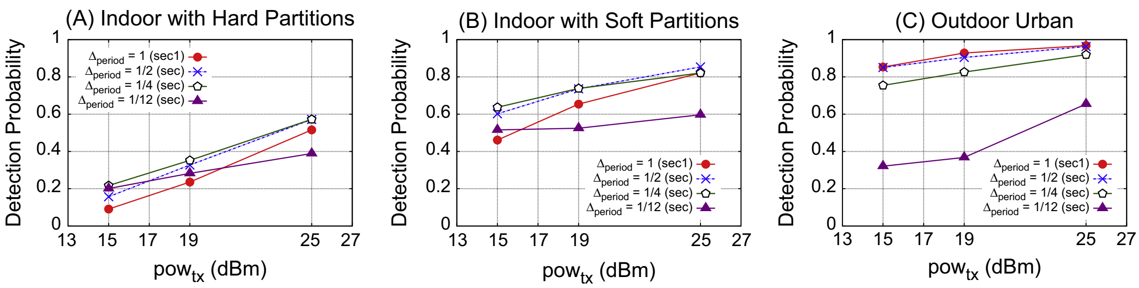

Fig. 4. Impact of increasing pow_{tx} on the detection probability in different environments.

- *Decreasing Δ_{period} for a given pow_{tx}.* For a fixed value of pow_{tx}, decreasing Δ_{period} could have different impacts on the detection probability depending on the environment (see Fig. 5). For instance, in indoor environments, decreasing Δ_{period} down to a certain value (e.g., 1/4 s in *indoor with hard partitions*) can increase the detection probability. However, below this value the detection probability starts to decrease due to the increase in collisions and interferences. In *outdoor urban*, decreasing Δ_{period} generally decreases the detection probability.

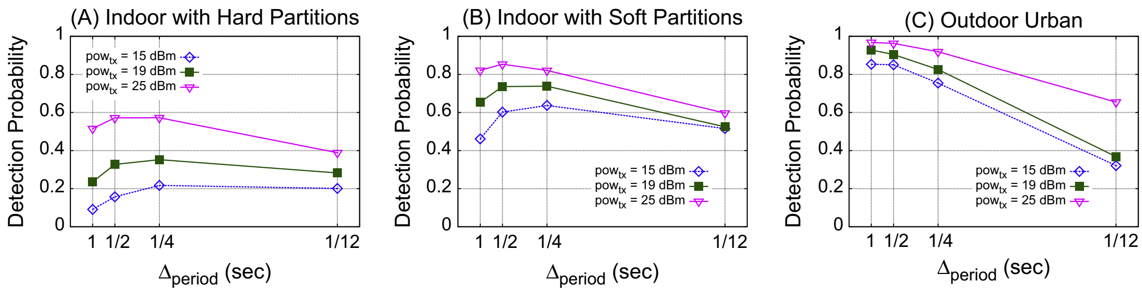

Fig. 5. Impact of decreasing Δ_{period} on the detection probability in different environments.

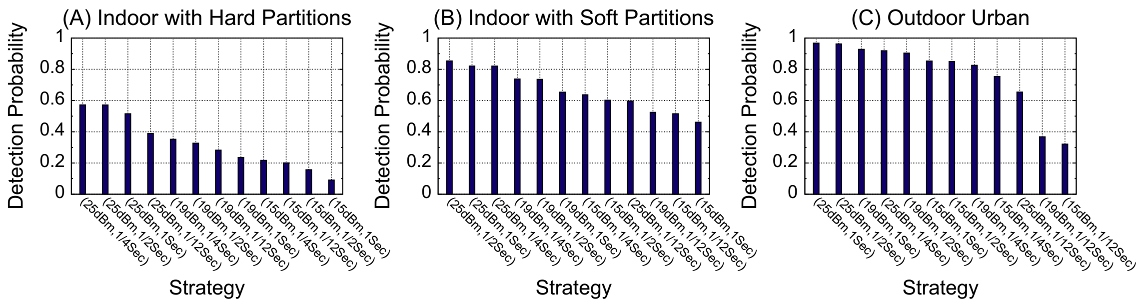

Fig. 6. Strategies ranked in the descending order with respect to their corresponding detection probability in different environments.

This is because the packets are less attenuated by the environment (compared to indoor environments) and a good percentage of them arrive to the receiver's physical layer with acceptable SINR. Thus, decreasing Δ_{period} only increases collisions and prevents the reception of the acceptable packets.

4.3.4. The most effective strategy

Fig. 6, depicts the strategies ranked in descending order with respect to their resulting detection probability in different environments. As shown, the most effective strategy is not the same in all environments. More precisely, Strategy (25 dBm, 1/4 s), Strategy (25 dBm, 1/2 s) and Strategy (25 dBm, 1 s) are respectively the most effective strategies in *indoor with hard partitions*, *indoor with soft partitions* and *outdoor urban areas*. These results are justified by our observations in Section 4.3.3.

Based on the detection probability comparisons, for a given strategy s and environment e, we defined the following metrics:

- *Effectiveness rank*. This rank is out of 12 strategies in set S and is based on the ranking in Fig. 6, i.e., in descending order with respect to detection probability. The effectiveness rank of the most effective strategy is 1.
- *Effectiveness ratio*. This is the ratio of the detection probability of s to the detection probability of the most effective strategy in environment e. Informally speaking, it compares the effectiveness of s to the maximum effectiveness that can be achieved in environment e. The effectiveness ratio of the most effective strategy is 1.

Table 2, depicts the effectiveness ranks and ratios of the strategies in different environments. We use these metrics later in Section 6 when discussing the tradeoff between effectiveness and efficiency.

5. Evaluation of efficiency

Efficiency is measured by the inverse of energy consumption per process. Therefore, in this section we first define a model of energy consumption and then design an algorithm that, based on the model, calculates for each strategy the energy consumption per process. After describing the algorithm, we present the results and we discuss how a change in pow_{tx} or Δ_{period} can affect the energy consumption. Finally we compare the efficiency of the strategies and present the most efficient strategy.

5.1. Energy consumption model

Communication is the primary cause of energy consumption of a node executing the neighbor detection algorithm. Since 802.11 (or WiFi) communication is considered one of the main causes of the battery discharge in mobile devices [33,30], in this section we only consider the energy consumption of the 802.11a wireless network interface card (or WNIC). As already described in Section 3, WNIC is the component that implements the MAC and the physical layers of the OSI model in a mobile device. To calculate the energy consumption of the WNIC for each strategy, we first define its energy consumption model as below.

Power is defined as the amount of energy consumed per unit of time. It is known that a 802.11 WNIC exhibits

Table 2
Effectiveness rank and ratio of the strategies in different environments.

Environment		(15 dBm, 1 Sec)	(15 dBm, 1/2 Sec)	(15 dBm, 1/4 Sec)	(15 dBm, 1/12 Sec)	(19 dBm, 1 Sec)	(19 dBm, 1/2 Sec)	(19 dBm, 1/4 Sec)	(19 dBm, 1/12 Sec)	(25 dBm, 1 Sec)	(25 dBm, 1/2 Sec)	(25 dBm, 1/4 Sec)	(25 dBm, 1/12 Sec)
Indoor-hard	Rank	12	11	9	10	8	6	5	7	3	2	1	4
	Ratio	0.16	0.27	0.37	0.35	0.41	0.57	0.61	0.49	0.90	0.99	1	0.67
Indoor-soft	Rank	12	8	7	11	6	5	4	10	2	1	3	9
	Ratio	0.54	0.70	0.74	0.60	0.76	0.862	0.864	0.61	0.962	1	0.961	0.69
Outdoor-urban	Rank	6	7	9	12	3	5	8	11	1	2	4	10
	Ratio	0.88	0.87	0.77	0.33	0.95	0.93	0.85	0.38	1	0.99	0.94	0.67

(Column header: Strategy)

different power consumptions at different radio modes. Therefore, in order to define the energy consumption model of the WNIC, we should first identify its different radio modes.

In this paper, we assume that the WNIC does not use any power saving mechanism for the IEEE 802.11 distributed coordination function (DCF). In fact, according to the power saving mechanism of the 802.11 standard, the WNIC sleeps most of the time and wakes up periodically to check whether there are some packets that it should transmit or receive. During the sleep mode no transmission or reception is possible and the power consumption is extremely low. Thus, If a node has a packet to transmit, it should buffer the packet and waits until the next wake-up time [1]. In this paper, we do not use the power saving mechanism for two main reasons. First, waking up at the right moment requires the nodes to have synchronized clocks. Second, the power saving mechanism is known to perform poorly when the number of nodes is high. In fact, at the wake-up time, a sender node should first announce the list of the buffered packets to destinations. The announcement is performed by sending an *ad-hoc traffic indication map* (*ATIM*) packet. As the number of nodes increases, more ATIM transmissions could take place at the same time which result in more collisions and hence lower performance [43]. Although some researchers proposed new power saving mechanisms without some of these limitations (for instance, schemes that work with asynchronous clocks [25,55]), in this paper for simplicity's sake we do not use any of these mechanisms. We might consider their potential use in our future work.

Thus, we assume that the WNIC can only operate in one of three radio modes, namely, *transmit*, *receive* and *idle* [1]. As their names suggest, the transmit and the receive modes correspond respectively to the cases where the WNIC transmits or receives a packet. In general, power consumption in the transmit mode is different from power consumption in the receive mode, since different circuits are used in these modes [38]. In the idle mode, the WNIC is required to continuously sense the medium. Thus, intuitively the power consumption in the idle mode should be similar to the power consumption in the receive mode. The experimental results in [19] confirm this fact and show that the power consumption in the idle mode is only slightly different from the power consumption in the receive mode. Therefore, we assume that in the idle mode, the WNIC consumes the same amount of power as in the receive mode. This assumption results in two general radio modes:

- *Active mode*. This mode is characterized by power consumption pow_{active} and corresponds to the cases where WNIC is in the transmit mode.
- *Passive mode*. This mode is characterized by power consumption $pow_{passive}$ and corresponds to the cases where the WNIC is either in the idle or the receive modes.

Knowing these two general radio modes, the energy consumption model can be specified as follows. Let T_{total} be the time duration for which the energy consumption of WNIC is defined. Then, T_{total} can be split into T_{active} and $T_{passive}$, which denote respectively the duration spent in the active and passive modes. Since power consumption of each mode is known, the energy consumption of each mode can be calculated separately. Let E_{total} denote the total energy consumption of WNIC during T_{total}, then E_{total} can be found by summing $E_{passive}$ and E_{active}, where $E_{passive}$ and E_{active} denote respectively the energy consumption in the active and passive modes [14].

5.2. Energy consumption calculation algorithm

To obtain the energy consumption of the WNIC for different strategies, we devise an algorithm called *energy consumption calculation* algorithm. It is based on the energy consumption model described in Section 5.1. It calculates the active, passive and total energy consumption of the WNIC for a given strategy and for a given time duration. The algorithm calculates the energy consumption independently of the environment. The main reason is that the time that the WNIC of a node (executing the neighbor detection algorithm) spends in each radio mode is independent of the environment attenuation.

Moreover, in order for the algorithm to characterize accurately the energy consumption of a current smartphone's WNIC, we use the power consumption specifications of Broadcom's BCM 4328 WNIC which is also used as a reference in [39] to devise power consumption equations.

The algorithm has three input parameters: the transmission power pow_{tx}, the time duration Δ_{period} (these two parameters form Strategy $(pow_{tx}, \Delta_{period})$), and the time duration T_{total} for which the energy consumption is calculated. The algorithm has three output parameters: the energy consumption in the active mode E_{active}, the energy consumption in the passive mode $E_{passive}$ and the total energy consumption E_{total}.

The algorithm has two main steps:

1. For $m \in \{active, passive\}$, perform the following:
   (a) Calculate pow_m, where pow_m is the power consumption of m mode.
   (b) Calculate T_m, where T_m is the amount of time out of T_{total} that is spent in m mode.
   (c) Set $E_m = pow_m \times T_m$, where E_m is the amount of energy consumed in m mode.
2. Set $E_{total} = \sum_{m \in \{active, passive\}} E_m$, where E_{total} is the total energy consumed during T_{total} and return $E_{active}, E_{passive}$ and E_{total} as output.

In the following, we describe two Substeps 1a and 1b.

(1a) Calculate pow_m: depending on the value of m, there exist two cases:

- If m is equal to *active*, pow_{active} should be calculated. To do so, we use Eq. (1) where all quantities are in milliwatts (mWs). This equation was introduced in [39]. As discussed in Section 5.1, pow_{active} is the power consumed by WNIC while transmitting packets. Eq. (1) shows how pow_{active} can be defined in terms of transmission power pow_{tx}.

$$pow_{active} = 305 + \frac{pow_{tx}}{0.02 \times 5^{[\frac{2}{3} \times \log_{10}(pow_{tx})]}} \tag{1}$$

  The first term of the equation, i.e., 305 mW, represents the common power consumed by the circuitry independent of pow_{tx}. This value is obtained based on Broadcom's BCM 4328 WNIC specifications [13]. The second term of the equation represents the total power consumed by the RF power amplifier of the WNIC, which is active during transmissions.
- If m is equal to *passive*, $pow_{passive}$ should be calculated. In [39], power consumed by WNIC during the reception is assumed to be always equal to 295 mW based on Broadcom's BCM 4328 WINC specifications. This assumption seems to be correct since the experimental results in [18] show that the amount of power consumed by the WNIC during the reception is not influenced by the transmission power pow_{tx}. Thus, we also adopt this assumption in our algorithm and set the value of $pow_{passive}$ to 295 mW.

(1b) Calculate T_m: depending on the value of m, there exist two cases:

- If m is equal to *active*, T_{active} should be calculated. Intuitively, T_{active} is the sum of transmission times of all packets that are transmitted by WNIC during T_{total}. Thus, let T_{tx} be the transmission time of a packet and n be the number of packets transmitted during T_{total}, T_{active} can be defined as:

$$T_{active} = n \times T_{tx} = \left\lfloor \frac{T_{total}}{\Delta_{period}} \right\rfloor \times 0.000728 \tag{2}$$

  where all quantities are in seconds. Note that in Eq. (2), n and T_{tx} are respectively replaced by $\left\lfloor \frac{T_{total}}{\Delta_{period}} \right\rfloor$ and 0.000728. Below we explain how these values are obtained.
  We find the value of T_{tx} using Eq. (3). This Equation was introduced in [38]. It calculates the transmission time of an *802.11a* data frame in seconds given its payload size $\mathcal{L}$ in bytes and the Bytes-per-Symbol information ($\mathcal{B}pS$) which is itself a function of data rate $\mathcal{R}$.

$$T_{tx} = 0.00002 + \left\lceil \frac{30.75 + \mathcal{L}}{\mathcal{B}pS(\mathcal{R})} \right\rceil \times 0.000004 \tag{3}$$

  In our case, i.e., with data rate of 6 Mbps, $\mathcal{B}pS = 3$. Since $\mathcal{L} = 500$ bytes, we find $T_{tx} = 0.000728$ s.
  To find the value of n, we should find the number of messages transmitted during T_{total}. According to the neighbor detection algorithm, a message is sent by the application layer every Δ_{period}. So, by setting the value of n to $\left\lfloor \frac{T_{total}}{\Delta_{period}} \right\rfloor$, we make three assumptions. First, we assume that one application layer message results in one MAC frame. This assumption conforms to our simulation study in Section 4.[7] Second, we assume that no message is dropped by the interface queue. This

[7] In real life applications, this assumption is not always true. However, to prevent a high number of packet collisions and network congestion, the *hello* packets are usually small packets resulting in few MAC frames. In addition, T_{active} is a linear function of number of transmitted frames. Therefore, this assumption does not influence our observations in Section 5.3.

assumption conforms to our system model. Third, we assume that the amount of time spent by a message between its sending by the application layer until the end of its transmission from the physical layer is smaller than Δ_{period}. This assumption is also reasonable considering the values of Δ_{period} of the studied strategies and is also confirmed by our simulation study described in Section 4. Based on these assumptions, we know that a message sent by the application layer is always transmitted to the channel before the sending of the next message by the application layer. Hence, all messages sent by the application layer during T_{total} are transmitted during $T_{total} + \epsilon$, with ϵ being negligible (ϵ accounts for limit conditions where the last message is sent very close to the end of the measurement period).

- If m is equal to *passive*, $T_{passive}$ should be calculated. $T_{passive}$ is the part of T_{total} that the WNIC spends in the reception or the idle modes, i.e., it is the part of T_{total} that WSN does not spend in the active mode. Therefore, $T_{passive}$ is calculated using Eq. (4) where T_{active} is replaced by Eq. (2). Note that all quantities in Eq. (4) are in seconds.

$$\begin{aligned} T_{passive} &= T_{total} - T_{active} = T_{total} - (n \times T_{tx}) \\ &= T_{total} - \left(\left\lfloor \frac{T_{total}}{\Delta_{period}} \right\rfloor \times 0.000728 \right) \end{aligned} \tag{4}$$

5.3. Results

Using the energy consumption evaluation algorithm, we calculated $E_{active}, E_{passive}$ and E_{total} for each strategy during $T_{total} = \Delta_{neighborhood} = 4$ s. Based on these calculations, we reached some observations described below. Note that these observations are valid for any value of T_{total}since the output energies of the energy consumption algorithm are linear functions of T_{total}.

- *Regardless of strategy, the majority of E_{total} consists of $E_{passive}$*. The results of different strategies show that, on average, 98.9% of E_{total} consists of $E_{passive}$ and only 1.08% consists of E_{active}. Roughly speaking, the reason is that regardless of the used strategy, the WNIC spends much more time in the passive mode than in the active mode. In fact, among all tested strategies, Strategy (25 dBm, 1/12 s) is the one that has the maximum value for E_{active} and at the same time the minimum value for $E_{passive}$.[8] When we compare T_{active} and pow_{active} of this strategy with its corresponding $T_{passive}$ and $pow_{passive}$, we realize that its pow_{active} is 4.69 times greater than $pow_{passive}$ (recall that $pow_{passive}$ is always equal to 295 mW). However, its corresponding T_{active} is still about 113.46 times smaller than its $T_{passive}$. That is why its resulting E_{active} is still much smaller than its $E_{passive}$.
- *E_{active} has a high variation between different strategies while $E_{passive}$ remains more or less constant*. Our results reveal that by changing strategy, E_{active} varies a lot whereas $E_{passive}$ tends to remain more or less the same. In fact, let E^{max}_{active} and E^{min}_{active} denote respectively the maximum and minimum of E_{active} for the tested strategies, and $E^{max}_{passive}$ and $E^{min}_{passive}$ denote respectively the maximum and minimum of $E_{passive}$ for the tested strategies. Then, E^{max}_{active} is 26.7821 times greater than E^{min}_{active}, whereas $E^{max}_{passive}$ is more or less equal to $E^{min}_{passive}$ since $E^{max}_{passive}/E^{min}_{passive} = 1.0080$ (see Fig. 7).
- *Variation of E_{total} between different strategies is mainly due to variation of E_{active}*. This observation is the direct consequence of the observation described in the previous point. Intuitively, since E_{active} varies a lot between strategies while $E_{passive}$ remains more or less constant, the variation of E_{total} is mainly due to E_{active}. For instance, among the tested strategies, Strategy (25 dBm, 1/12 s) is the one that maximizes E_{total} and Strategy (15 dBm, 1 s) is the one that minimizes E_{total}. Thus, if we change the strategy from (25 dBm, 1/12 s) to (15 dBm, 1 s) we save 3.05% in E_{total}. This is the result of saving 3.82% in E_{active} and at the same time losing 0.77% in $E_{passive}$.

Based on these observations, we directly consider the active energy consumption instead of the total energy consumption while measuring the efficiency of each strategy. As shown in Fig. 7, under $T_{total} = \Delta_{neighborhood} = 4$ s, the amount of energy saved by applying the strategy with minimum E_{active} instead of other strategies is very low (i.e., at most 47 milliJoules (mJ)). However, since E_{active} is a linear function of time, this amount becomes more and more significant as T_{total} increases (see Fig. 8). In other words, choosing a strategy that minimizes the active energy consumption will not result in saving much energy in the short-term but in the long-term. Note that, henceforth, we use the term *energy consumption* instead of *active energy consumption* for simplicity's sake.

5.3.1. Impact of changing pow_{tx} and Δ_{period} on energy consumption

In this section, we discuss the impact of increasing pow_{tx} and decreasing Δ_{period} on energy consumption.

- *Increasing pow_{tx} for a given Δ_{period}*. According to the energy consumption calculation algorithm, for a given Δ_{period}, when we increase pow_{tx}, we in fact, increase the power consumption in the active mode or pow_{active} which in its turn increases the energy consumption. Thus, pow_{active} is about 621.22 mW, 822.13 mW and 1386.48 mW when transmitting with pow_{tx} of 15 dBm, 19 dBm and 25 dBm, respectively. This means that at a given Δ_{period}, energy consumption is slightly increased (by 1.3 times) when we increase pow_{tx} from 15 dBm to 19 dBm and is almost doubled when we increase pow_{tx} from 15 dBm or 19 dBm to 25 dBm (see Fig. 9A).

[8] A strategy can maximize E_{active} if it can maximize at the same time T_{active} and pow_{active}. Roughly speaking, it is the strategy that results in the maximum number of transmitted packets and the maximum transmission power. On the other hand, a strategy can maximize $E_{passive}$ if it can just maximize $T_{passive}$, since $pow_{passive}$ is constant and not a function of transmission power. In other words, all strategies that result in the minimum number of transmitted packets, maximize $E_{passive}$. A similar reasoning can be applied to find the strategies that minimize E_{active} and $E_{passive}$.

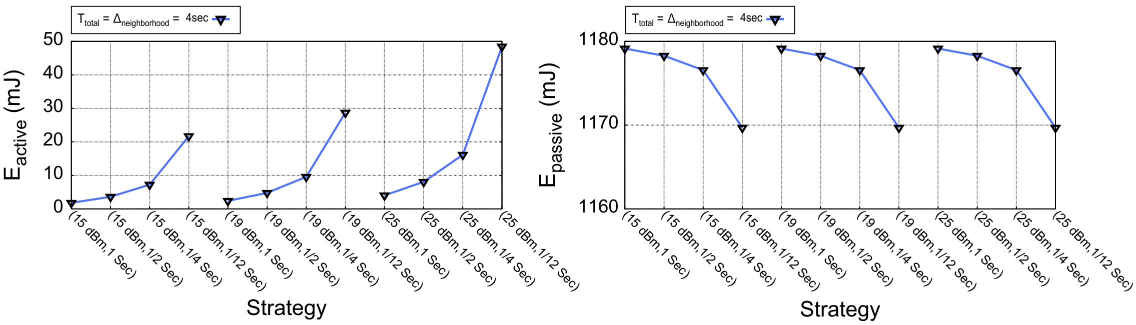

Fig. 7. Variation of E_{active} and $E_{passive}$ over different strategies when $T_{total} = \Delta_{neighborhood} = 4$ s.

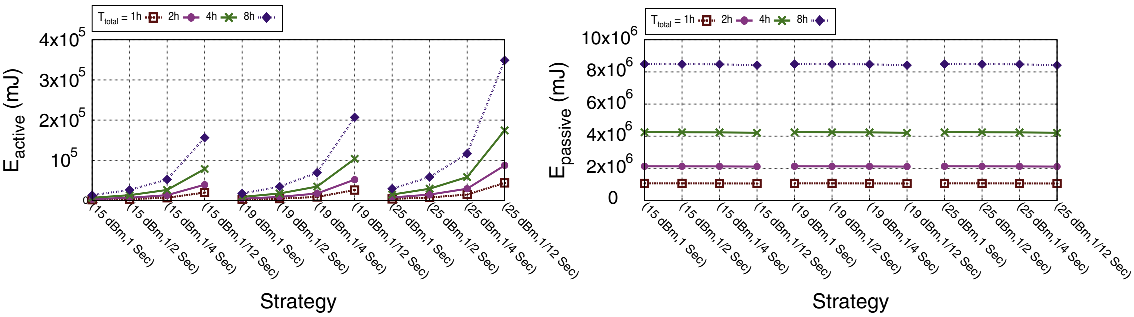

Fig. 8. Variation of E_{active} and $E_{passive}$ over different strategies for high values of T_{total} (different hours).

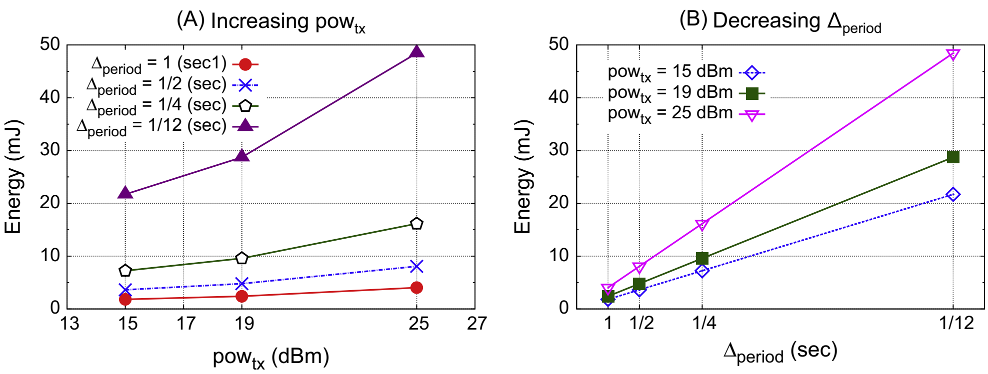

Fig. 9. Impact of increasing pow_{tx} and decreasing Δ_{period} on the energy consumption.

- *Decreasing Δ_{period} for a given pow_{tx}.* According to the energy consumption calculation algorithm, E_{active} is a linear function of $1/\Delta_{period}$. Thus, for a given pow_{tx}, decreasing Δ_{period} increases the energy consumption in a linear manner. This increase is at least twice. Recall that by increasing pow_{tx}, the energy consumption is at most doubled. Thereby, the impact of decreasing Δ_{period} on energy consumption is usually more significant than the impact of increasing pow_{tx} (see Fig. 9B).

5.3.2. The most efficient strategy

Fig. 10, depicts the strategies ranked in ascending order with respect to their corresponding energy consumption. As discussed in Section 5.3.1, compared to increasing pow_{tx}, decreasing Δ_{period} (which accordingly increases the number of transmitted packets) has generally a more significant impact on the increase of energy consumption. That is why for instance, Strategy (15 dBm, 1/2 s) consumes more energy than Strategy (19 dBm, 1 s) in spite of the fact that Strategy (15 dBm, 1/2 s) uses a lower transmission power compared to Strategy (19 dBm, 1 s).

As depicted in Fig. 10, among all tested strategies, Strategy (15 dBm, 1 s) and Strategy (25 dBm, 1/12 s) consume the minimum and maximum amount of energy, respectively. Therefore, the most efficient strategy is Strategy (15 dBm, 1 s). Note that the most efficient strategy is the

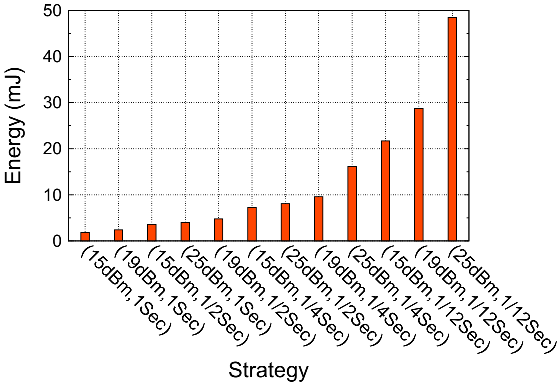

Fig. 10. Strategies ranked in the ascending order with respect to their corresponding energy consumption.

same in all environments since energy consumption is calculated independently of environment.

Based on the energy consumption comparisons, for a given strategy s, we defined the two following metrics:

- *Efficiency rank*. This rank is out of 12 strategies in set S and is based on the ranking in Fig. 10 i.e., in ascending order with respect to energy consumption. Efficiency rank of the most efficient strategy is equal to 1.
- *Efficiency ratio*. This is the ratio of energy consumption of the most efficient strategy to the energy consumption of s. Informally speaking, it compares the efficiency of s to the maximum efficiency that can be achieved. The efficiency ratio of the most efficient strategy is 1.

Table 3, depicts the efficiency ranks and ratios of the strategies. We use these metrics later in Section 6 when discussing the tradeoff between effectiveness and efficiency.

6. Effectiveness-efficiency tradeoff

In this section, we first compare the results of effectiveness and efficiency evaluations to find the strategy that maximizes both effectiveness and efficiency for each environment. We show that there is a conflict between effectiveness and efficiency. Hence, such a strategy does not exist in any environment. We then propose an approach to make a tradeoff between effectiveness and efficiency and we find the tradeoff strategy for each environment. Finally, to show how good the tradeoff strategy is, for each environment, we compare its effectiveness and its efficiency to the maximum effectiveness and efficiency that can be achieved in that environment.

6.1. Conflict

A summary of the results of the effectiveness and efficiency evaluations can be found in Tables 4,5.[9] As depicted in Table 4, the most effective strategy and the most efficient strategy are not the same in any environment.

In fact, there is a conflict when we try to maximize both effectiveness and efficiency. The reason is that, most of the time, a change in pow_{tx} or Δ_{period} does not influence effectiveness and efficiency similarly and in some cases it results in opposite behaviors (see Table 5). For instance, consider pow_{tx}. Based on the evaluations, we know that regardless of environment, increasing pow_{tx} increases effectiveness, whereas it decreases efficiency. That is why the most effective strategy has the highest pow_{tx} (i.e., 25 dBm) in all environments whereas the most efficient strategy has the lowest pow_{tx} (i.e., 15 dBm). Also, regarding Δ_{period}, we know that decreasing Δ_{period} (down to some degree) increases effectiveness in indoor environments and decreases effectiveness as the environment becomes less obstructed. At the same time, decreasing Δ_{period} decreases efficiency. Thereby, Δ_{period} of the most effective strategy increases from 1/4 s to 1/2 s and then to 1 s as we change the environment from *indoor with hard partitions* to *indoor with soft partitions* and then to *outdoor urban areas*, respectively. On the other hand, the most efficient strategy has the highest Δ_{period} (i.e., 1 s).

Note however that the conflict becomes less severe as the environment becomes less obstructed. For instance, as we change the environment from the *indoor with hard partitions* to *indoor with soft partitions* and then to *outdoor urban areas*, the efficiency ratio of the most effective strategy increases from 0.11 to 0.22 and then to 0.44, respectively. As a result, its efficiency rank increases from 9th to 7th and then to 4th (see Table 4). In fact, as already discussed, the impact that a change in Δ_{period} has on effectiveness and efficiency becomes similar as we move from indoor environments to outdoor. Therefore, Δ_{period} of the most effective strategy becomes higher (and closer to Δ_{period} of the most efficient strategy) in less obstructed environments.

Similarly, the most efficient strategy becomes more effective as the environment becomes less obstructed. More precisely, as depicted in Table 4, as we change the environment from the *indoor with hard partitions* to *indoor with soft partitions* and then to *outdoor urban areas*, the effectiveness ratio of the most efficient strategy increases from 0.16 to 0.54 and then to 0.88, respectively. As a result, its effectiveness rank increases from 12th position to 6th position (see Table 4). In fact, compared to other strategies, the most efficient strategy has the lowest pow_{tx} (i.e., 15 dBm) and the highest Δ_{period} (i.e., 1 s). As already discussed, the positive impact of a high Δ_{period} on effectiveness becomes more significant as the environment becomes less obstructed. At the same time, based on the results in Section 4.3.3, we know that the negative impact that a low pow_{tx} has on effectiveness becomes less considerable as the environment becomes less obstructed.

6.2. Approach to make the tradeoff

Up to this point, our goal was to find, for each environment, the strategy that has both maximum effectiveness and efficiency among all other strategies. In order to achieve our goal, we evaluated the effectiveness and the

[9] Tables 4,5 summarize the results of different sections of this paper. In this section, we do not discuss the results in rows or columns corresponding to *tradeoff strategy* or *BCR*.

Table 3
Efficiency rank and ratio of the strategies.

	Strategy											
	(15 dBm, 1 Sec)	(15 dBm, 1/2 Sec)	(15 dBm, 1/4 Sec)	(15 dBm, 1/12 Sec)	(19 dBm, 1 Sec)	(19 dBm, 1/2 Sec)	(19 dBm, 1/4 Sec)	(19 dBm, 1/12 Sec)	(25 dBm, 1 Sec)	(25 dBm, 1/2 Sec)	(25 dBm, 1/4 Sec)	(25 dBm, 1/12 Sec)
Rank	1	3	6	10	2	5	8	11	4	7	9	12
Ratio	1	0.5	0.25	0.08	0.75	0.37	0.18	0.06	0.44	0.22	0.11	0.03

Table 4
Comparison of the most effective strategy, the most efficient strategy and the tradeoff strategy in different environments.

Environment	Strategy	Detection Probability	Energy Consumption (mJ)	BCR (1/mJ)	Effectiveness Ratio	Efficiency Ratio	Effectiveness Rank	Efficiency Rank
Indoor-hard	**The Most Effective Strategy** (25 dBm, 1/4 Sec)	0.5722	16.15	0.035	1	0.11	1	9
	The Most Efficient Strategy (15 dBm, 1 Sec)	0.0910	1.809	0.050	0.16	1	12	1
	The Tradeoff Strategy (25 dBm, 1 Sec)	0.5157	4.037	0.127	0.90	0.44	3	4
Indoor-soft	**The Most Effective Strategy** (25 dBm, 1/2 Sec)	0.8536	8.07	0.105	1	0.22	1	7
	The Most Efficient Strategy (15 dBm, 1 Sec)	0.4612	1.809	0.254	0.54	1	12	1
	The Tradeoff Strategy (19 dBm, 1 Sec)	0.6539	2.39	0.273	0.76	0.75	6	2
Outdoor-urban	**The Most Effective Strategy** (25 dBm, 1 Sec)	0.9679	4.04	0.239	1	0.44	1	4
	The Most Efficient Strategy (15 dBm, 1 Sec)	0.8532	1.809	0.471	0.88	1	6	1
	The Tradeoff Strategy (15 dBm, 1 Sec)	0.8532	1.809	0.471	0.88	1	6	1

efficiency of each strategy separately. However, due to the conflict described in Section 6.1, such strategy does not exist. Moreover, in a given environment, the strategy that has the highest effectiveness usually has a low efficiency and the strategy that has the highest efficiency is not always very effective. Therefore, we need to find a tradeoff strategy, that is a strategy that on one hand does not consume a lot of energy and on the other hand results in a high (but maybe not the highest) detection probability compared to other strategies.

The main idea for making the tradeoff comes from the concept of *cost–benefit analysis* in the field of economy [9]. Thus, we identify the resulting detection probability of a strategy as its benefit and its resulting energy consumption as its cost. Then, for each strategy, the *benefit–cost ratio* (*BCR*) is calculated. Finally, the strategy that

Table 5
Impact of changing pow_{tx} or Δ_{period} on effectiveness, efficiency and BCR in different environments.

Environment	Increasing pow_{tx}			Decreasing Δ_{period}		
	Impact on effectiveness	Impact on efficiency	Impact on BCR	Impact on effectiveness	Impact on efficiency	Impact on BCR
Indoor-hard	Improves	Deteriorates	improves only under $\Delta_{period} = 1$ second.	Generally improves if Δ_{period} is decreased down to 1/4 second.	Deteriorates	Deteriorates
Indoor-soft	Improves	Deteriorates	improves only under $\Delta_{period} = 1$ second and when pow_{tx} is increased up to 19 dBm.	Generally improves if Δ_{period} is decreased down to 1/2 second.	Deteriorates	Deteriorates
Outdoor-urbant	Improves	Deteriorates	Deteriorates	Deteriorates	Deteriorates	Deteriorates

results in the highest ratio is chosen as the tradeoff strategy. This means that the tradeoff strategy is the one that makes the best use of energy for neighbor detection.

6.3. Benefit–Cost Ratio (BCR)

Let e be the environment in which we apply a strategy s. Also, let $Pr_{detect}(s, e)$ denote the neighbor detection probability obtained by applying s in e and $E(s)$ denote the energy consumption due to applying s. Then, BCR achieved by applying s in e or $BCR(s, e)$ is defined as:

$$BCR(s,e) = \frac{Pr_{detect}(s,e)}{E(s)} \tag{5}$$

6.4. Impact of changing pow_{tx} and Δ_{period} on BCR

We use Figs. 11 and 12 to discuss the impact of changing pow_{tx} and Δ_{period} on BCR. Intuitively, changing the value of pow_{tx} or Δ_{period} can only improve BCR if the resulting gain in the detection probability is more significant than the loss of energy. More precisely, suppose that a change in the value of one of these parameters, changes the detection probability by a factor of x and the energy consumption by a factor of y. Then, BCR is increased if $x/y > 1$. If $x/y = 1$, BCR does not change. Finally, if $x/y < 1$, BCR decreases. In the following, we use this observation to interpret the results.

- *Increasing pow_{tx} for a given Δ_{period}.* We know that for a fixed value of Δ_{period}, increasing pow_{tx} increases the detection probability in all environments. It also increases the energy consumption. However as shown in Fig. 11, it can have different impacts on the value of BCR depending on the environment. For instance, under $\Delta_{period} = 1$ s, increasing pow_{tx} from 15 dBm to 25 dBm increases BCR in the *indoor with hard partitions* environment and decreases BCR in the *outdoor-urban* environment. In fact, in both cases the energy consumption is increased by a factor of 2.23 (recall that the energy consumption is independent of environment). However, in the *indoor with hard partitions* environment the detection probability is increased by a factor of 5.66, whereas, in the *outdoor urban* environment, it is only increased by a factor of 1.13.
  To summarize the results we can say that increasing pow_{tx} improves BCR only in indoor environments, under high values of Δ_{period} and for certain ranges of pow_{tx} (see Fig. 11). In fact, as already discussed in Section 4.3.3, although in all environments increasing pow_{tx} improves the detection probability, in general, the improvement becomes less significant as the environment becomes less obstructed. Accordingly, increasing pow_{tx} can lead to a decrease in BCR in less obstructed environments.
- *Decreasing Δ_{period} for a given pow_{tx}.* For a fixed value of pow_{tx}, decreasing Δ_{period} decreases the value of BCR regardless of the environment (see Fig. 12). In fact, we know that decreasing Δ_{period} increases the energy consumption considerably. We also know that decreasing Δ_{period} can have different impacts on the detection probability depending on the environment. In particular, in indoor environments, decreasing Δ_{period} down to some value (such as 1/4 s in *indoor with hard partitions* environment) can improve the detection probability. However, in all these cases the loss in energy consumption is so significant that it mitigates the potential gain in detection probability.

6.5. The tradeoff strategy

Fig. 13, shows the strategies ranked in descending order with respect to their corresponding BCR in different environments. As already discussed in Section 6.4, decreasing Δ_{period} decreases BCR in all cases. That is why the tradeoff strategy has $\Delta_{period} = 1$ s in all environments. Moreover, increasing pow_{tx} improves BCR only when the environment is obstructed and as the environment becomes less obstructed it can even decrease BCR. That is why the tradeoff strategy in *indoor with hard partitions*, *indoor with soft partitions* and *outdoor urban* environments has respectively pow_{tx} of 25 dBm, 19 dBm and 15 dBm.

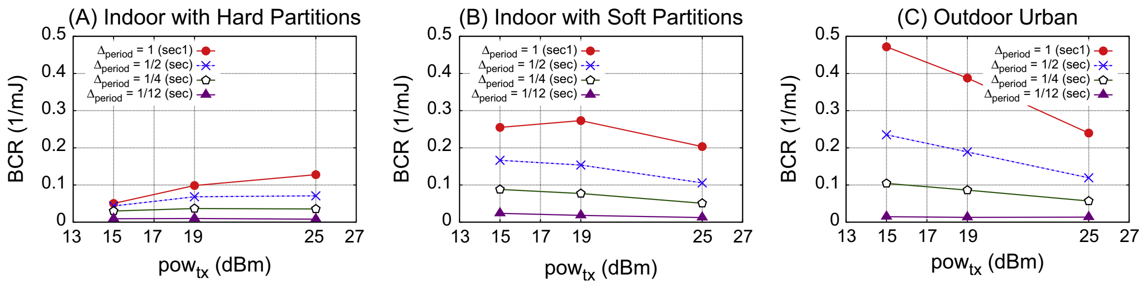

Fig. 11. Impact of increasing pow_{tx} on BCR in different environments.

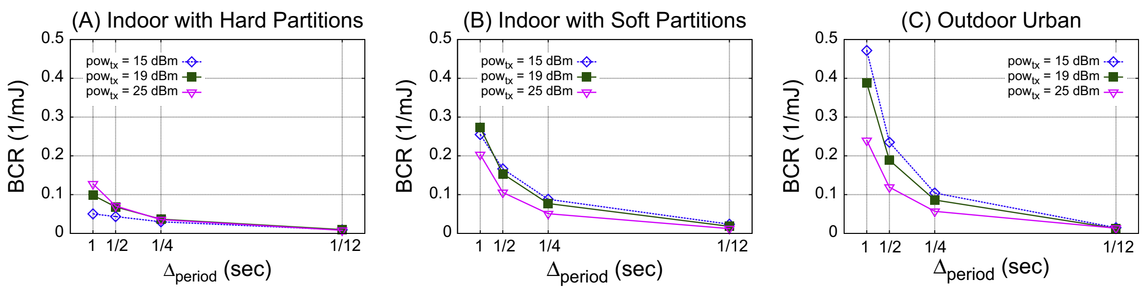

Fig. 12. Impact of decreasing Δ_{period} on BCR in different environments.

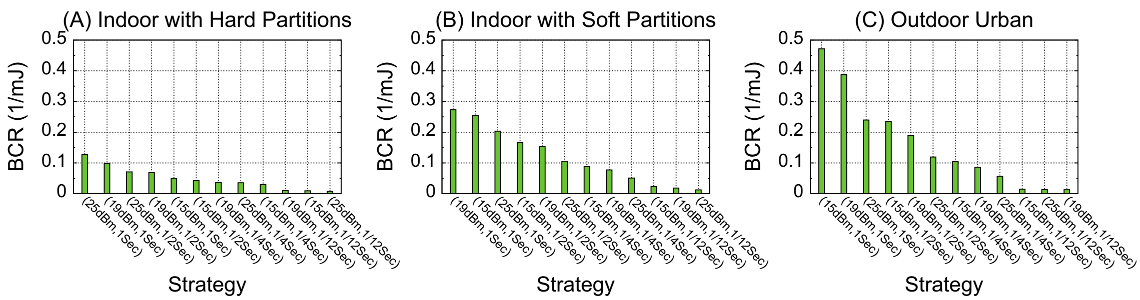

Fig. 13. Strategies ranked in the descending order with respect to their corresponding BCR in different environments.

6.6. How effective and efficient is the tradeoff strategy?

A summary of the results of BCR evaluation can be found in Tables 4 and 5. As shown in Table 4, in all environments, the effectiveness ratio of the tradeoff strategy is high and close to 1. More precisely, the effectiveness ratio of the tradeoff strategy is equal to 0.90, 0.76 and 0.88 in *indoor with hard partitions*, *indoor with soft partitions* and *outdoor urban areas*, respectively. Accordingly, the tradeoff strategy has a relatively good effectiveness rank i.e., 3rd, 6th and 6th in *indoor with hard partitions*, *indoor with soft partitions* and *outdoor urban areas*, respectively.

Moreover, the efficiency ratio of the tradeoff strategy is equal to 0.44, 0.75 and 1 in *indoor with hard partitions*, *indoor with soft partitions* and *outdoor urban areas*, respectively (see Table 4). In fact, Δ_{period} of the tradeoff strategy is equal to 1 s in all environments and we know that a high value of Δ_{period} results in low energy consumption. At the same time, pow_{tx} of the tradeoff strategy decreases as the environment becomes less obstructed which causes the decrease of its energy consumption in less obstructed environment and increase of its efficiency ratio. Thereby, the tradeoff strategy has a relatively good efficiency rank i.e., 4th, 2nd and 1st in *indoor with hard partitions*, *indoor with soft partitions* and *outdoor urban areas*, respectively. Note that in the *outdoor urban* environment, the tradeoff strategy is the same as the most efficient strategy, and therefore it has the lowest energy consumption compared to other strategies.

7. Related work

The existing studies on transmit power control in ad hoc networks [27], for the most part, consider the unicast communications and therefore are not relevant to our work. Also, most of the papers which study the problem of reliable broadcast in MANETs, consider one-shot broadcast and not periodic broadcast [21]. For these reasons, in this section we only discuss the works performed in the two fields which we believe are the closests to our work, i.e., *enhancements of the hello protocol*; and *beacon broadcast for VANET safety applications*.

7.1. Enhancements of the hello protocol

Neighbor detection in ad hoc networks is usually studied as a building block for applications such as routing, leader election, group management and localization. The neighbor detection algorithm introduced in this paper is inspired by the *basic hello protocol* first described in *Open Shortest Path First* (*OSPF*) routing protocol [35]. It works as follows. Nodes periodically send *hello* messages to announce their presence to close nodes, and maintain a neighborhood table. The sending frequency is denoted by f_{hello}. Thus, in the case of our neighbor detection algorithm $f_{hello} = 1/\Delta_{period}$.

In the literature, there exist several works that attempt to improve the basic hello protocol performance especially with regards to energy consumption. These works can be classified into two categories described below.

The first category consists of the algorithms that aim at minimizing the energy consumption by keeping the nodes in the sleep radio mode most of the time and thus, reaching neighbor detection with as few transmission and/or reception attempts as possible [34,42,51,17,48,26,6]. The majority of the enhancements of the hello protocol belongs to this category. However, as described in Section 5.1, we do not consider the sleep radio mode in this paper. Therefore, we do not discuss the works in this category here.

The second category (which is more relevant to our work) consists of the algorithms that aim at minimizing the energy consumption of neighbor detection by (regularly) adapting f_{hello} and/or the transmission power to the network changes [22,24,23]. For instance, in the protocol proposed in [22], the value of f_{hello} is adapted based on link connectivity. More precisely, the authors define two metrics: Time to Link Failure (TLF) and Time Without link Changes (TWC). Thus, each node regularly evaluates these metrics and compares them with some given thresholds. Then, based on the comparisons, the node can switch f_{hello} from a high value (f_{high}) to a low value (f_{low}) and vice versa. The drawback of this approach is that the thresholds may need to evolve over time and finding the correct thresholds is not obvious.

The protocols in [24,23] adapt f_{hello} to the variations of node speed. They are based on the existence of an optimal *hello* frequency denoted by f_{opt}. The equation for f_{opt} was first introduced in [47] where it is calculated based on the relative speed of nodes. In [24], the authors proposed a Turnover based Adaptive hello Protocol (TAP). According to TAP, each node evaluates the changes of its neighborhood table periodically (i.e., before sending each *hello*) and calculates the turnover. The turnover (also dented by r) is, in fact, the ratio between the number of new neighbors (i.e., nodes detected during the last period) and the current total number of neighbors. Once the turnover r is calculated, it is compared to r_{opt} where r_{opt} is the turnover corresponding to f_{opt}. If $r < r_{opt}$, this means that f_{hello} is too high and there are not enough changes in the table. Therefore, f_{hello} should be decreased. On the contrary, if $r > r_{opt}$, this means that f_{hello} is too low and that there are too many changes. Therefore, f_{hello} should be increased. In this way, TAP keeps the value of f_{hello} always close to f_{opt}. In [23] a Turnover based Frequency and transmission Power Adaptation algorithm (TFPA) is presented. TFPA applies a similar (but slightly improved) approach as TAP to dynamically adapt f_{hello}. Moreover, TFPA calculates an optimal value for transmission range based on the energy consumption model presented in [20]. Then, it dynamically adapts the transmission power so that the resulting transmission range remains close to the optimal value. Compared to TAP, TFPA has a lower energy consumption since it dynamically adapts both f_{hello} and the transmission power. However, both protocols have assumptions which are less realistic compared to the assumptions in our work. More precisely, they consider the unit disk graph and a deterministic radio propagation model whereas we use a probabilistic radio propagation model. Moreover, the energy consumption model assumed by TFPA, corresponds to sensor nodes and does not realistically reflect the energy consumption of WNICs used in today's mobile devices whereas we model the energy consumption based on specifications of current smartphones. Finally, both protocols use f_{opt} which is basically determined by the relative speed of nodes. As a result, they essentially adapt f_{hello} according to the speed changes and neglect the other factors such as the environment attenuation or congestions. In our work, on the other hand, we take into account the environment attenuations as well as interferences and collisions and neglect the relative speed of the nodes. In fact this choice is justified by the fact that in our case the devices are usually used by pedestrians. Hence, they have a slow movement especially during the neighborhood time of 4 s.

7.2. Beacon broadcast for VANET safety applications

In vehicular ad hoc networks (VANETs), safety applications aim at minimizing accident levels. One type of safety messages are beacons. A beacon usually contains the vehicle's position, speed, direction, etc and is periodically broadcast in a single-hop manner. By using beacons, safety applications gain knowledge of the surroundings and can prevent dangerous situations for the drivers. The safety applications usually require the beacon delivery up to a certain distance and have some guarantees for message reliability and latency. Thus, many papers study the impact of parameters such as transmission power, packet generation rate or packet size on fulfillment of applications' guarantees [54,45,46,32].

For instance, in [45] authors present a simulation-based study in ns-2.28 to analyze the impact of transmission power and packet generation rate on the reception of beacon messages. Authors consider 1800 nodes uniformly distributed in a circular map. All nodes broadcast messages with common transmission power and packet generation rate. The broadcast reception is only studied for messages of senders located at 40 m from a receiver which is located at the center of the map. Regarding the impact of transmission power on broadcast reception, authors state that the transmission power should be strong enough to resist the interferences but not so strong as to increase the load on the medium. Regarding the impact of packet generation rate on broadcast reception, they state that increasing packet generation rate can increase the number of received packets significantly, as long as the channel busy time (or the time ratio a node determines the channel as busy) has not reached its maximum. There are several differences between this study and ours: firstly, this study is performed for VANETs. Thus, the 802.11 physical layer parameters are set to the specific values of the *802.11p* standard whereas we use the values of the 802.11a standard. Secondly, this study uses the *Nakagami* radio propagation model known to model signal attenuation in VANETs, whereas we use LNS to model obstructed urban areas. Thirdly, this study uses different metrics (such as channel busy time) than our metrics to interpret the results. Finally, this study does not consider the impact of transmission power or packet generation rate on the energy consumption.

In [46], authors show that periodic beacon transmission can result in the channel saturation which in turn, causes a high number of packet collisions and low reception rates. They also show that simply increasing the packet generation rate or the transmission power can exacerbate the channel conditions. Therefore, they propose an algorithm called Distributed Fair Power Adjustment for Vehicular environments (D-FPAV). The algorithm limits the beaconing load on the channel below a predefined threshold while ensuring a high probability of beacon delivery at close distances from the sender. Similarly to the work in [45], this work also considers the *802.11p* physical and MAC layers and the *Nakagami* radio propagation model and thus has a different system model compared to our work. However, the basic idea behind the D-FPAV algorithm is to fix the packet generation rate at the minimum required by safety applications, and to adjust the transmission power of beacons in case of congestion. This means that the way that the D-FPAV algorithm adapts the transmission power and the packet generation rate somehow resembles the way that our tradeoff strategy is adapted in different environments. In fact, for the tradeoff strategy we keep Δ_{period} at its highest value and we decrease the transmission power in less obstructed environments i.e., where the collisions and interferences are high.

8. Conclusion

To the best of our knowledge, this is the first paper that studies the impact of transmission power and broadcast interval on effectiveness and efficiency of neighbor detection for MANETs in different urban environments. Our results can be used as a basis to design adaptive neighbor detection algorithms for urban environments. Such algorithms can adapt the transmission power and broadcast interval based on environment and application guarantees on effectiveness and efficiency. To deploy such algorithms on a smartphone, one can use lightweight sensing services such as the one introduced in [56] which can detect the indoor/outdoor environment in a fast, accurate, and efficient manner.

Relying on a realistic simulation-based study, we showed that the most effective strategy is not the same as the most efficient strategy in any environment. In fact, in all environments, there is a conflict between effectiveness and efficiency such that the most effective strategy is usually not very efficient and the most efficient strategy is not always very effective. However, we showed that the conflict becomes less severe as the environment becomes less obstructed. When discussing our results, we also described how a change in transmission power and broadcast interval can influence the effectiveness and efficiency. We then proposed an approach to make a tradeoff between effectiveness and efficiency. Accordingly, we identified the tradeoff strategy in each environment and we showed that it has a relatively good effectiveness and efficiency compared to other strategies.

The conclusions drawn in this paper could still be more realistic if our evaluations could be performed on a real prototype. In fact, we are currently developing, *ManetLab*, a modular and configurable software framework for creating and running testbeds to evaluate MANET-specific protocols [50]. Thus, as a potential future work, we consider to deploy the neighbor detection algorithm on ManetLab and compare the results of our evaluations with the ones performed using the ManetLab.

There are also some issues which remain open and we might consider as future work. For instance, the possibility to detect nodes using an underlying multi-hop network should be investigated. Moreover, in this paper we did not consider the sleep radio mode for 802.11 communication. Since for energy consumption we only took into account the active energy, it seems that our results could still be valid for the case when the sleep radio mode is enabled. Thus, this issue can also be investigated as future work.

Acknowledgment

This research is partially funded by the Swiss National Science Foundation in the context of Project 200021-140762.

References

[1] IEEE 802.11: Wireless LAN MAC and Physical Layer Specifications, 1999.
[2] IEEE std 802.11a/D7.0-1999, part11: wireless LAN medium access control (MAC) and physical layer (PHY) specifications: high speed physical layer in the 5 GHz band, 1999.
[3] Official IEEE 802.11 working group project timelines, 2014. <http://grouper.ieee.org/groups/802/11/Reports/802.11_Timelines.htm>.
[4] Apple iPhone 4S Teardown. <http://www.techinsights.com/teardowns/apple-iphone-4s-teardown/>.
[5] F. Bai, A. Helmy, A Survey of Mobility Modeling and Analysis in Wireless Ad Hoc Networks, Wireless Ad Hoc and Sensor Networks, Springer, 2006.
[6] M. Bakht, M. Trower, R. Kravets, Searchlight: helping mobile devices find their neighbors, in: ACM SIGOPS Oper. Syst., vol. 45, no. 3, 2012, pp. 71–76.
[7] S. Basagni, M. Conti, S. Giordano, I. Stojmenovic, Mobile Ad Hoc Networking: The Cutting Edge Directions., Wiley-IEEE Press, 2013.
[8] Best iPhone bluetooth games. <http://iphone.mob.org/genre/multipleer/>.
[9] N.E. Boardman, Cost-Benefit Analysis: Concepts and Practice, third ed., Upper Saddle River, NJ, Prentice Hall, 2006.
[10] B. Bostanipour, B. Garbinato, A. Holzer, Spotcast – a communication abstraction for proximity-based mobile applications, in: Proc. IEEE NCA'12, 2012, pp. 121–129.
[11] B. Bostanipour, B. Garbinato, Improving neighbor detection for proximity-based mobile applications, in: Proc. IEEE NCA'13, 2013, pp. 177–182.

[12] B. Bostanipour, B. Garbinato, Using virtual mobile nodes for neighbor detection in proximity-based mobile applications, in: Proc. IEEE NCA'14, 2014, pp. 9–16.
[13] Broadcom. <http://www.broadcom.com/>.
[14] M.M. Carvalho, C.B. Margi, K. Obraczka, J.J. Garcia-Luna-Aceves, Modeling energy consumption in single-hop IEEE 802.11 ad hoc networks, in: Proc. IEEE ICCCN'04, 2004, pp. 367–372.
[15] Q. Chen, F. Schmidt-Eisenlohr, D. Jiang, M. Torrent-Moreno, L. Delgrossi, H. Hartenstein, Overhaul of IEEE 802.11 modeling and simulation in NS-2, in: Proc. ACM MSWiM'07, 2007, pp. 159–168.
[16] R. Coté, G.E. Harrington, Life safety code handbook (NFPA 101), 2009.
[17] P. Dutta, D. Culler, Practical asynchronous neighbor discovery and rendezvous for mobile sensing applications, in: Proc. ACM SenSys'08, 2008.
[18] J.-P. Ebert, B. Burns, A. Wolisz, A trace-based approach for determining the energy consumption of a WLAN network interface, in: Proc. of European Wireless, 2002, pp. 230–236.
[19] L.M. Feeney, M. Nilsson, Investigating the energy consumption of a wireless network interface in an ad hoc networking environment, in: Proc. IEEE INFOCOM'01, 2001, pp. 1548–1557.
[20] E. Fleury, D. Simplot-Ryl, Réseaux de capteurs, Hermes Science – Lavoisier, 2009.
[21] B. Garbinato, A. Holzer, F. Vessaz, Context-aware broadcasting approaches in mobile ad hoc networks, In Comput. Netw. 54 (7) (2010) 1210–1228.
[22] C. Gomez, A. Cuevas, J. Paradells, AHR: a two-state adaptive mechanism for link connectivity maintenance in AODV, in: Proc. ACM REALMAN'06, 2006.
[23] D. He, N. Mitton, D. Simplot-Ryl, An energy efficient adaptive HELLO algorithm for mobile ad hoc networks, in: Proc. ACM MSWiM'13, 2013, pp. 65–72.
[24] F. Ingelrest, N. Mitton, D. Simplot-Ryl, A turnover based adaptive hello protocol for mobile ad hoc and sensor networks, in: Proc. MASCOTS'07, 2007, pp. 9–14.
[25] J.-R. Jiang, Y.-C. Tseng, C.-S. Hsu, T.-H. Lai, Quorum-based asynchronous power-saving protocols for IEEE 802.11 ad hoc networks, ACM Mob. Netw. Appl. 10 (1–2) (2005) 169–181.
[26] A. Kandhalu, K. Lakshmanan, R.R. Rajkumar, U-connect: a low-latency energy-efficient asynchronous neighbor discovery protocol, in: IPSN'10, 2010, pp. 350–361.
[27] V. Kawadia, P. Kumar, Principles and protocols for power control in wireless ad hoc networks, IEEE JSAC 23 (1) (2005) 76–88.
[28] V. Kelly, New IEEE 802.11ac specification driven by evolving market need for higher, multi-user throughput in wireless LANs, IEEE, 2014.
[29] J. Lee, W. Kim, S-J Lee, D. Jo, J. Ryu, T. Kwon, Y. Choi, An experimental study on the capture effect in 802.11a networks, in: Proc. ACM WinTECH'07, 2007, pp. 19–26.
[30] J. Li, J. Xiao, J.W.-K. Hong, R. Boutaba, Application-centric Wi-Fi energy management on smart phone, in: Proc. IEEE APNOMS'12, 2012, pp. 1–8.
[31] LoKast. <http://www.lokast.com/>.
[32] X. Ma, J. Zhang, T. Wu, Reliability analysis of one-hop safety-critical broadcast services in VANETs, IEEE Trans. Vehicular Tech. 60 (8) (2011) 3933–3946.
[33] J. Manweiler, R.R. Choudhury, Avoiding the rush hours: WiFi energy management via traffic isolation, IEEE Trans. Mobile Comput. 11 (5) (2012) 739–752.
[34] M.J. McGlynn, S.A. Borbash, Birthday protocols for low energy deployment and flexible neighbor discovery in ad hoc wireless networks, in: Proc. ACM MobiHoc'01, 2001, pp. 137–145.
[35] J. Moy, OSPF – Open Shortest Path First, RFC 1583, 1994.
[36] T. Muetze, P. Stuedi, F. Kuhn, G. Alonso, Understanding radio irregularity in wireless networks, in: Proc. IEEE SECON'08, 2008, pp. 82–90.
[37] ns-2.35 (released on November 4, 2011). <http://www.isi.edu/nsnam/ns/>.
[38] D. Qiao, S. Choi, A. Jain, K.G. Shin, Miser: an optimal low-energy transmission strategy for IEEE 802.11a/h, in: Proc. ACM MobiCom'03, 2003, pp. 161–175.
[39] K. Ramachandran, R. Kokku, H. Zhang, M. Gruteser, Symphony: synchronous two-phase rate and power control in 802.11 WLANs, IEEE/ACM Trans. Netw. 18 (4) (2010) 1289–1302.
[40] T.S. Rappaport, Wireless Communications: Principles and Practice, second ed., Prentice Hall, 2002.
[41] M. Raya, J.P. Hubaux, The security of vehicular ad hoc networks, in: Proc. ACM SASN'05, 2005, pp. 11–21.
[42] G. Sawchuk, G. Alonso, E. Kranakis, P. Widmayer, Randomized protocols for node discovery in ad hoc multichannel networks, in: Proc. Ad Hoc Now, 2003.
[43] J. Schiller, Mobile Communications, second ed., Addison-Wesley, England, 2003.
[44] F. Schmidt-Eisenlohr, Interference in vehicle-to-vehicle communication networks – analysis, modeling, simulation and assessment, PhD thesis, Karlsruhe Institute of Technology (KIT), 2010.
[45] M. Torrent-Moreno, S. Corroy, F. Schmidt-Eisenlohr, H. Hartenstein, IEEE 802.11-based one-hop broadcast communications: understanding transmission success and failure under different radio propagation environments, in: Proc. ACM MSWiM'06, 2006, pp. 68–77.
[46] M. Torrent-Moreno, J. Mittag, P. Santi, H. Hartenstein, Vehicle-to-vehicle communication: fair transmit power control for safety-critical information, IEEE Trans. Vehicular Tech. 58 (2009) 3684–3703.
[47] A. Troël, Prise en compte de la mobilité dans les interactions de proximité entre terminaux à profils hétérogènes, PhD thesis, Université de Rennes, 2004.
[48] Y.-C. Tseng, C.-S. Hsu, T.-Y. Hsieh, Power-saving protocols for IEEE 802.11-based multi-hop ad hoc networks, in: Proc. IEEE INFOCOM'02, 2002.
[49] W. Vandenberghe, I. Moerman, P. Demeester, On the feasibility of utilizing smartphones for vehicular ad hoc networking, in: Proc. IEEE ITST'11, 2011, pp. 246–251.
[50] François Vessaz, Benoît Garbinato, Arielle Moro, Adrian Holzer, Developing, deploying and evaluating protocols with ManetLab, in: Proc. NETYS'13, 2013, pp. 89–104.
[51] G. Wattenhofer, G. Alonso, E. Kranakis, P. Widmayer, Randomized protocols for node discovery in ad hoc, single broadcast channel networks, in: Proc. IPDPS'03, 2003.
[52] Waze. <https://www.waze.com/en/>.
[53] WhosHere. <http://whoshere.net/>.
[54] Q. Xu, T. Mak, J. Ko, R. Sengupta, Vehicle-to-vehicle safety messaging in DSRC, in: Proc. ACM VANET'04, 2004, pp. 19–28.
[55] R. Zheng, J.C. Hou, L. Sha, Asynchronous wakeup for ad hoc networks, in: Proc. ACM MobiHoc'03, 2003, pp. 35–45.
[56] P. Zhou, Y. Zheng, Z. Li, M. Li, G. Shen, IODetector: a generic service for indoor outdoor detection, in: Proc. ACM SenSys'12, 2012, pp 113–126.

Behnaz Bostanipour is a PhD student in computer science at the University of Lausanne in the Distributed Object Programming (DOP) Lab under the supervision of Prof. Garbinato. Before joining the DOP Lab, she obtained a BSc. degree and a MSc. degree in Communication Systems from Swiss Federal Institute of Technology of Lausanne (EPFL). Her research interests focus on mobile ad hoc networks and distributed computing. In particular, she studies and designs new abstractions and algorithms for proximity-based mobile applications running on mobile devices and smartphones.

Benoît Garbinato is a professor in computer science at the University of Lausanne since 2004, where he leads the Distributed Object Programming (DOP) Lab. In the nineties, he contributed to the emerging research trend on separation concerns and protocol composition in fault-tolerant distributed systems, as part of his Ph.D. thesis. He then worked in the industry, first for the research lab of UBS in Zurich (Ubi-lab), where he lead the software engineering group, and later for Sun Microsystems professional services, as senior software architect. Since his return to the academic world, his research and teaching activities focus on the design and implementation of adequate programming abstractions for emerging distributed architectures, such as pervasive and mobile systems. He has over 50 publications in international conferences and journals, and is member of the ACM and the IEEE societies.