Improving Neighbor Detection for Proximity-Based Mobile Applications

Behnaz Bostanipour   Benoît Garbinato

Distributed Object Programming Laboratory
University of Lausanne
CH-1015 Lausanne, Switzerland
Email: {behnaz.bostanipour,benoit.garbinato}@unil.ch

***Abstract*—In this paper, we consider the problem of improving the detection of a device by another device in mobile ad hoc networks, given a maximum amount of time that they remain in proximity of each other. 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. 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 be detected as a neighbor. Thus, we study the effect of parameters that we believe could influence the detection probability, i.e., *the transmission power* and *the time interval between two consecutive broadcasts*. More precisely, for each environment, we determine when a change in the value of each of these parameters could lead to an improvement of the neighbor detection and when it hurts. Our experiments show that there exists no unique combination of values of these parameters that maximizes the detection probability in all environments. Accordingly, for each environment, we present the combination that maximizes the detection probability in that environment.**

***Keywords*-Neighbor Detection, Proximity-Based Broadcast, MANET, Distributed Systems, IEEE 802.11, Smartphone;**

I. 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. These applications enable a user to interact with others in a defined range and for a given time duration e.g., for social networking [11], [12], gaming [13] or driving [14].

Discovering who is nearby is a basic requirement of various PBM applications. In a simple usage scenario of social networking applications such as WhosHere [11] or LoKast [12], 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 this paper, we consider the following problem: *how could the detection probability of a device by another device be increased, given a maximum amount of time that they remain in proximity of each other?* To address this problem, we perform a realistic simulation-based study in a single-hop mobile ad hoc network (MANET) using ns-2.35 [15] network simulator.

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.

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. For physical and mac layer, we use an implementation of *IEEE 802.11a* in ns-2.35. Thus, we study the effect of parameters that we believe could influence the detection probability, i.e., *the transmission power* and *the time interval between two consecutive broadcasts*. In performing the simulations, we are particularly interested to answer the following questions:

- *For each environment, when could a change in the value of any of the above mentioned parameters lead to an improvement of the neighbor detection, or on the contrary, when does it deteriorate the detection?*
- *Is there a unique combination of the above mentioned parameters that could maximize the neighbor detection probability in all environments?* If it is not the case, *in each environment, what is the combination that maximizes the value of the detection probability?*

The remainder of the paper is as follows. In Section II, we describe our system model. In Section III, we present our goal and the approach to achieve it. We also describe our simulation setup. In Section IV, we present our simulation results; moreover, we define two packet dropping metrics which are useful to interpret the results. At the end of this section, for each environment, we present the combination of the studied parameters that maximizes the detection probability. Finally, we discuss related work in Section V before concluding in Section VI with a perspective on future work. .

II. System Model

In this section, we present the system model, and whenever necessary, we describe how its elements are mapped in our simulations and the reasons behind our modeling choices.

A. Processes

We consider a mobile ad-hoc network (MANET) consisting of a finite set of n processes $P = \{p_1, p_2, ..., p_n\}$. 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.

B. 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.

C. 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]. 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 [16] and uses the similar modulation schemes and coding rates for broadcast as 802.11a. Thus, it is useful to have the results which could be also valid for this standard. For the implementation of 802.11a in ns-2.35, we use the one presented in [1]. This implementation includes a fully revised and enhanced architecture for physical and mac layers to improve the drawbacks of the 802.11 default support in ns-2.

D. 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 [3]. 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 (σ = 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. For our simulations, we use the implementation of LNS model in ns-2.35 and we consider a distinct pair of (β, σ) values for each environment (see Table I). These pairs are chosen based on the measurements in the literature [3].

TABLE I: Values of LNS Parameters for each Environment.

Environment	$\boldsymbol{\beta}$	$\boldsymbol{\sigma}$ (dB)
Indoor-hard partitions	5.5	7
Indoor-soft partitions	5	9.6
Outdoor-urban	4	5.5

E. 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.

III. Evaluation Approach and Setup

Let $\Delta_{neighborhood}$ be the maximum amount of time that a node p_i remains continuously (i.e., without any leaving and re-entering) within the detection range R of a node p_j, then, for each environment, our goal is to find the pair (pow_{tx}, Δ_{period}) that maximizes the detection probability of p_i by p_j. Thus, we call a pair (pow_{tx}, Δ_{period}) a *strategy*.

A. Approach

To achieve our goal, we test a set of predefined strategies in each environment by performing simulations. Thus, for each strategy (pow_{tx}, Δ_{period}), we initialize the *neighbor detection algorithm* at all nodes with pow_{tx} and Δ_{period}. Since we want to set the value of $\Delta_{neighborhood}$ to be 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 RNs. 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 $[0, \Delta_{neighborhood}]$.

B. Simulation Setup

We consider a square of 100m width filled with 1000 static nodes located using a uniform random distribution. RNs are nodes located at the distance less than or equal to 5m from the center of the square. This is because the nodes that are close to the center, usually experience the maximum radio interference, which can lead to the worst case of neighbor detection. The total number of nodes is chosen after studying the occupancy load factors of urban surfaces [4]. Transmission power pow_{tx} can take a value of 15 dBm, 19 dBm or 25 dBm. The first two values correspond to the common smartphone specifications, whereas the last value presents the possible performance gain of more powerful radio transmitters [5], [17]. The time interval between two consecutive broadcasts or Δ_{period} can take a value of 1 second, 1/2 second, 1/4 second or 1/12 second. In a simulation, all nodes have the same idealized transmission range[1] 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 30m. This value is chosen such that even using our lowest pow_{tx} choice, the detection range is less than the idealized transmission range.

We set $\Delta_{neighborhood}$ equal to 4 seconds for all simulations. We use the default values of the 802.11a implementation in ns-2.35 for physical and mac layers, however, we disable both *preamble* and *frame* capture features. For data rate, we consider 6 Mbps that uses 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. Packet size is set to 500 bytes. If messages are generated while previously generated messages are still not transmitted, the new messages are stored in an interface queue that is capable of storing up to 100 packets.

We consider all combinations of the above values for pow_{tx} and Δ_{period} and for each combination (or strategy), we perform five simulations with five different pairs of seeds, i.e., 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 randomness of topologies. Thereby, for each simulation, we have a different topology (with different RNs) and a LNS model which is seeded differently. Then, in order to obtain the output for the corresponding strategy, we take the average of five simulations' outputs. 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 $t \in [0, 4]$ seconds; thus, the output for a strategy is the average (over all RNs in the five simulations) of the neighbor detection probability at time t for all $t \in [0, 4]$ seconds. Henceforth, we use the term *neighbor detection probability* instead of the *average neighbor detection probability* for simplicity's sake.

IV. Evaluation Results

Our preliminary simulation results show a good detection probability (at least 0.8) for nodes situated at a maximum distance of 15m from a RN, in all environments. Therefore, in this section we only discuss the detection of nodes situated at a distance between 15m to 30m from a RN.

Fig. 1 depicts, as an example, the results for strategy (15 dBm, 1/4 Sec) in different environments. As shown in the figure and already described in Section III-A, 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*.

Intuitively, in a given environment increasing pow_{tx} and decreasing Δ_{period} (which increments the total number of sent *hello*s) should lead to the best strategy (i.e., the strategy that maximizes the detection probability in that environment). But 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. 3–A to Fig. 3–C), whereas decreasing Δ_{period} for a given pow_{tx} might result in unpredictable behaviors (see Fig. 3–D to Fig. 3–F). 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 how they can be used to interpret the results in one particular environment, i.e., *indoor with hard partitions*. Then, we present our general observations based on the interpretation of the results of all environments using our metrics.

A. Packet Dropping Metrics

Before defining our metrics, we present an overview of the packet dropping mechanism by 802.11 physical layer. Thus, 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*. This

[1] *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).

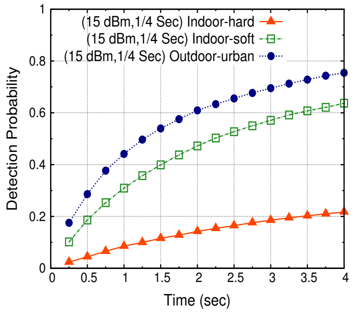


Fig. 1: Same strategy in different environments

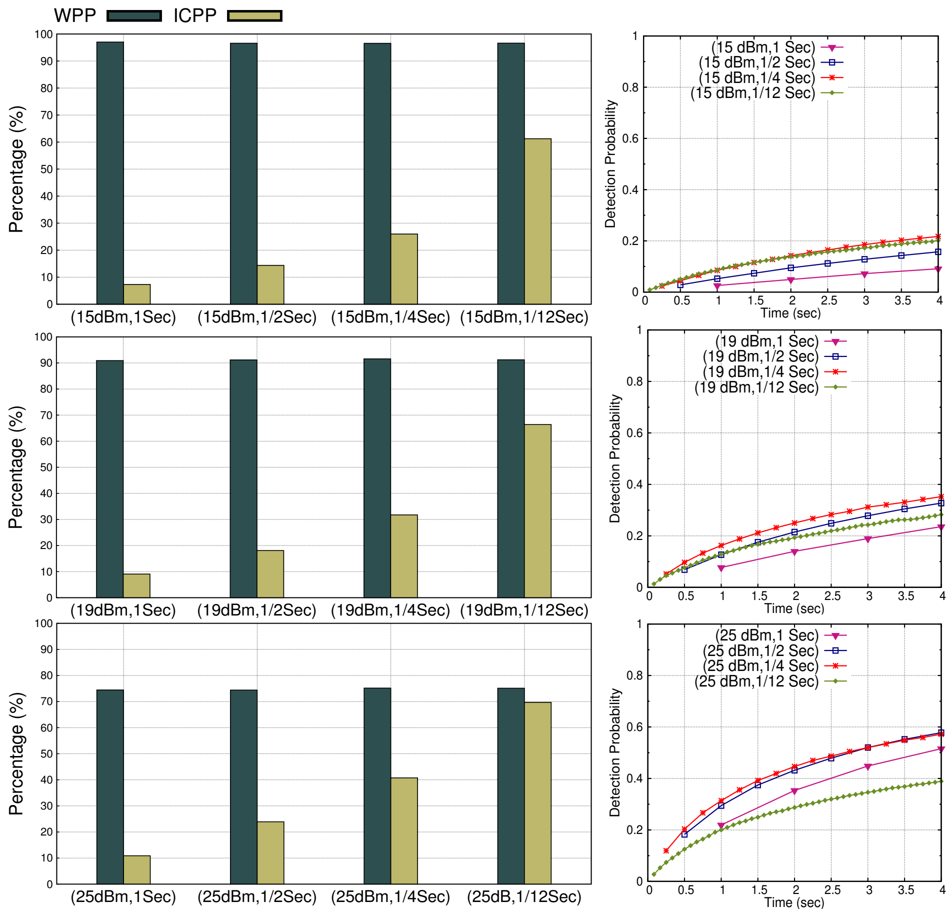
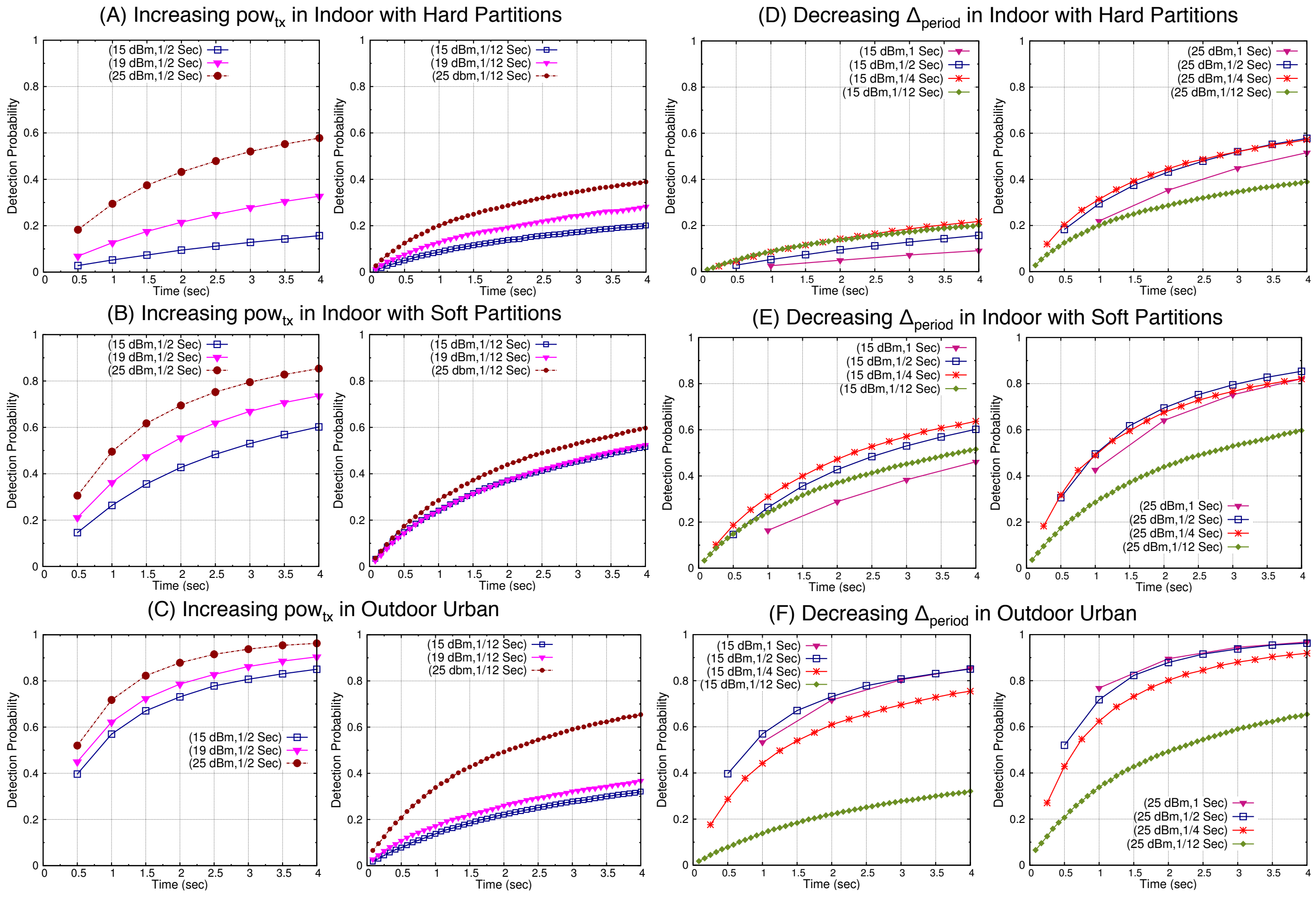


Fig. 2: Interpreting the results of indoor-hard partitions using packet dropping metrics


(A) Increasing pow_{tx} in Indoor with Hard Partitions

(D) Decreasing Δ_{period} in Indoor with Hard Partitions

(B) Increasing pow_{tx} in Indoor with Soft Partitions

(E) Decreasing Δ_{period} in Indoor with Soft Partitions

(C) Increasing pow_{tx} in Outdoor Urban

(F) Decreasing Δ_{period} in Outdoor Urban

Fig. 3: Effect of increasing pow_{tx} and decreasing Δ_{period} on the detection probability in different environments

threshold is defined by the modulation scheme and the coding rate.[2] If there is only one sender, the noise is equal to the thermal noise of the system. 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 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,[3] 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 15m to 30m 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 15m to 30m through a simulation. By 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 are 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 Δ_{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 (around 1% at maximum), so we simply ignore them.

B. Metrics-based Interpretation of the Results

We now interpret the results for *indoor with hard partitions* environment using our defined metrics. In this way, 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. 2 and Fig. 3–A. In particular, in Fig. 2, on the left, the values of the defined metrics for different combinations of pow_{tx} and Δ_{period} in *indoor with hard partitions* environment are presented and on the right, the corresponding detection probabilities for combinations of pow_{tx} and Δ_{period} are presented.

1) Increasing pow_{tx} for a given Δ_{period}: As shown in Fig. 2, as we increase pow_{tx} from 15 to 19 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. 3–A). However, when we increase pow_{tx} from 15dBm to 19dBm the detection probability improves differently under different values of Δ_{period}. For instance as shown in Fig. 3–A, when Δ_{period}=1/12 second, increasing pow_{tx} from 15 dBm to 19 dBm does not improve the detection probability as much as it improves under Δ_{period}=1/2 second. 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.

2) Decreasing Δ_{period} for a given pow_{tx}: As depicted in Fig. 2, 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 effect in the presence of different values of WPP. For instance, as shown in Fig. 2, 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. 2, with pow_{tx}=15 dBm, when Δ_{period}=1/12 second, the detection probability is greater than the detection probability at Δ_{period}=1 second and almost equal to the detection probability at Δ_{period}=1/4 second. However, when the value of WPP is relatively low, collisions have more effect and can decrease the detection probability e.g., as shown in Fig. 2, when pow_{tx}=25 dBm the value of WPP=75%. In this case when Δ_{period}=1/12 second, the detection probability is even less than the detection probability at Δ_{period}=1 second.

C. General Observations

After interpreting the results of all environments by using our metrics, we reach the following general observations.

1) *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 cases (see Fig. 3–A to Fig. 3–C). Increasing pow_{tx} from 15 dBm to 19 dBm, improves the neighbor detection under high values of Δ_{period} (e.g., for 1/2 seconds), but under low values of Δ_{period} (e.g., for 1/12 seconds), it has less effect and can even lead to no improvement. For instance, as shown in Fig. 3–B, in *indoor with soft partitions* environment, under Δ_{period}=1/2, increasing power from 15 dBm to 19 dBm increases the detection probability from 0.6020 to 0.7359 (at second 4), whereas under Δ_{period}=1/12, increasing pow_{tx} from 15 dBm to 19 dBm has no influence on the detection probability. This is because under low values of Δ_{period}, number of collisions and interferences is relatively high.

[2] We only consider one SINR threshold, since we use the same modulation scheme (BPSK) and coding rate (1/2) for both preamble and data frame.

[3] We do not consider *the capture effect* implemented in some chipsets.

2) *Decreasing Δ_{period} for a given pow_{tx}*: for a fixed value of pow_{tx}, decreasing Δ_{period} could have different effects on the detection probability depending on the environment (see Fig. 3–D to Fig. 3–F). For instance, in indoor environments, decreasing Δ_{period} down to a certain value (e.g., 1/4 second 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. 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.

D. Best Strategies

Now that we have studied the influence of two parameters pow_{tx} and Δ_{period} on the neighbor detection probability in each environment, we can devise the best strategies. A best strategy is a pair (pow_{tx}, Δ_{period}) that maximizes the detection probability in a given environment. Table II presents the best strategy and its corresponding detection probability in each environment. Note that in Table II, the values of the neighbor detection probabilities are taken at second 4, since the neighbor detection probability is an increasing function of time.

TABLE II: The Best Strategy for each Environment

Environment	(pow_{tx}, Δ_{period})	**Detection Probability**
Indoor-hard partitions	(25 dBm, 1/4 Sec)	0.5722
Indoor-soft partitions	((25 dBm, 1/2 Sec)	0.8536
Outdoor-urban	(25 dBm, 1 Sec)	0.9679

V. Related Work

Neighbor detection in ad hoc networks is usually studied as a building block for applications such as routing, leader election, group management and localization. However, the closest works to our work seem to be performed in the context of single-hop periodic broadcast of messages for safety applications in vehicular ad hoc networks (VANETs). These safety applications usually require the message delivery up to a certain distance and have some guarantees for message reliability and latency. Thus, many papers study the effect of parameters such as transmission power, packet generation rate or packet size on fulfillment of applications' guarantees [7], [6], [8], [9]. For instance, in [6] authors present a simulation-based study in ns-2.28 to analyze the effect of transmission power and packet generation rate on the reception of single-hop broadcast of safety messages in VANETs. 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 40m from a receiver which is located at the center of the map. Regarding the effect of transmission power on broadcast reception, authors state that the transmission power should be strong enough to resist the interferences but not too strong to increase the load on the medium. Regarding the effect of packet generation rate on broadcast reception, they state that increasing packet generation rate can increase the amount 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. Finally, this study uses different metrics (such as channel busy time) than our metrics to interpret the results.

VI. Conclusion

In this paper, we attempted to find the best strategies for neighbor detection in context of PBM applications running in MANETs. For our study, we considered typical urban environments and we used a simulation-based approach to find the best strategy and its corresponding benefit (the detection probability) in each environment. An issue that could be investigated as future work is the cost of each strategy in terms of energy consumption. In fact, the problem of energy consumption in ad hoc networks is more complicated than it first appears and can include many aspects, e.g., while studying energy consumption, we should consider not only the cost of transmitting a packet, but also receiving or discarding it [10].

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

References

[1] Q. Chen, F. Schmidt-Eisenlohr, D. Jiang, M. Torrent-Moreno, L. Delgrossi, and H. Hartenstein, Overhaul of IEEE 802.11 modeling and simulation in NS-2, In *Proc. ACM MSWiM'07*, pp. 159–168, 2007.

[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] T. S. Rappaport, Wireless communications: principles and practice, 2nd Edition, Prentice Hall, 2002.

[4] R. Cote, G. Harrington, Life safety code handbook (NFPA 101), 2009.

[5] BCM4329 and BCM4330 chipsets specifications, Broadcom, 2012.

[6] M. Torrent-Moreno, S. Corroy, F. Schmidt-Eisenlohr, and 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*, pp. 68–77, 2006.

[7] Q. Xu, T. Mak, J. Ko, and R. Sengupta, Vehicle-to-vehicle safety messaging in DSRC, In *Proc. ACM VANET'04*, pp. 19–28, 2004.

[8] M. Torrent-Moreno, J. Mittag, P. Santi, and H. Hartenstein, Vehicle-to-Vehicle communication: fair transmit power control for safety-critical information, *IEEE Trans. Vehicular Tech.*, vol. 58, pp. 3684–3703, 2009.

[9] X. Ma, J. Zhang, and T. Wu, Reliability analysis of one-hop safety-critical broadcast services in VANETs, *IEEE Trans. Vehicular Tech.*, vol. 60, no.8, pp. 3933–3946, 2011.

[10] L. M. Feeney and M. Nilsson, Investigating the energy consumption of a wireless network interface in an ad hoc networking environment, In *Proc. IEEE INFOCOM'01*, pp. 1548–1557, 2001.

[11] WhosHere. http://en.wikipedia.org/wiki/WhosHere

[12] LoKast. http://www.lokast.com/

[13] Game Mobile. http://www.gamemobile.co.uk/bluetoothmobilegames

[14] Waze. http://en.wikipedia.org/wiki/Waze

[15] ns-2.35 (released on Nov. 4, 2011). http://www.isi.edu/nsnam/ns/

[16] Official IEEE 802.11 working group project timelines, 2013.

[17] W. Vandenberghe, I. Moerman, and P. Demeester, On the feasibility of utilizing smartphones for vehicular ad hoc networking, In *Proc. IEEE ITST'11*, pp. 246–251, 2011.