



Improved Mobility Modeling for Indoor Localization Applications

Nicolas Amiot, Mohamed Laaraiedh, Bernard Uguen

► To cite this version:

Nicolas Amiot, Mohamed Laaraiedh, Bernard Uguen. Improved Mobility Modeling for Indoor Localization Applications. Positioning, Navigation and Communication (WPNC), 2014 11th Workshop on, Mar 2014, Dresden, Germany. pp.1-6, 2014, <10.1109/WPNC.2014.6843298>. <hal-01060185>

HAL Id: hal-01060185

<https://hal-univ-rennes1.archives-ouvertes.fr/hal-01060185>

Submitted on 3 Sep 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Improved Mobility Modeling for Indoor Localization Applications

Nicolas Amiot[‡], Mohamed Laaraiedh^{*}, Bernard Uguen[‡]

[‡] University of Rennes 1, IETR, France

^{*} Higher School of Communications of Tunis, University of Carthage, Tunisia

Abstract—This paper presents a novel mobility model to perform realistic simulations of human movements and behaviors. The proposed model is based on discrete event simulation and graph theory. The proposed model is implemented in a wireless propagation simulator and used to evaluate various wireless network protocols including: propagation, localization and communication.

Index Terms—Mobility modeling, Localization, DES, Graph theory.

I. INTRODUCTION

Evaluation of wireless networks protocols and algorithms should be done under realistic conditions. One of these conditions is realistic movement of mobile users. In particular, evaluation of cooperative and dynamic localization techniques needs realistic mobility models. In [1] and [2] we presented an open source radio propagation simulator PyLayers [3] whose purpose is to help the design and the evaluation of localization algorithms in heterogeneous networks. To ensure accurate simulation and evaluation of propagation, localization and communication protocols and algorithms implemented in PyLayers, a mobility module is developed as part of this simulator. This module implements a novel human-like mobility model based on discrete event simulation (DES).

PyLayers is organized in different layers ensuring each a particular task. The *Layout* layer ensures the definition of the environment layout using different graphs. The *Mechanical* layer mainly implements mobility aspects and will be the subject of the paper. The *Propagation* layer provides an estimation of the transmission channel using a ray-tracing approach. The *Network* layer implements inter-agents communication protocols. A *Localization* layer is also developed to implement various localization algorithms [4], [5]. In the following will be presented the mobility layer and examples of interactions with other layers and applications (i.e. propagation, localization and communication).

II. STATE OF THE ART OF MOBILITY AND NETWORK SIMULATORS

A. Mobility modeling

Building a wireless propagation simulator able to take into account the mobility should be focused on the choice of the correct mobility model. According to [6], mobility models can be categorized in three types:

- Random models
- Models with geographic restriction

- Models with spatio-temporal dependency

Random models are widely used in Monte-Carlo simulations where the purpose is to introduce a realistic fluctuation of the channel allowing the design and test of communications algorithms or PHY/MAC design. If those solutions are very easy to implement, the resulting motion is generally not very realistic. Those random models are based on Brownian motion or random way point. Some also add a temporal dependency and make the actual velocity dependent on the previous computed velocities. One of the most used model, due to its relative statistical coherence with a real human mobility, is the Levy Flight random walk model [7]. In the Levy flight model, the steps are defined in terms of the step-lengths, which have a certain probability distribution (the directions of the steps being isotropic and random). Those random models are very convenient to take into consideration the global behavior of an agent movement, but they are not suited for addressing site-specific situation. New approaches have to be considered to fit with deterministic simulation of the channel propagation.

It is mandatory that the simulated agent positions be bounded in space. That is the aim of models with geographic restriction which limits the movement of the agents into a restricted simulation field. In indoor situations, it means that a building floor plan would bound the position of the agents and limit their movement to possible way paths (crossing the doors). Graph based models belong to this category by constraining the movement of an agent on the graph nodes [8]. As well, the Manhattan Mobility model [9] limits the movement of the nodes to roads between buildings. Those space-restricted models are very convenient to describe the movement of agents, but they fail to take into account inter-agent interactions which require a synchronization-like behavior mechanism.

For that purpose, spatio-temporal dependent models are introduced. Those models take care of the *material* environment and the *human* environment. Inter-agent avoidance or group mobility to a common target can be envisaged which may drastically increase the realism of the simulation [10]. That level of realism is actually the lowest level required to build a satisfying simulation of the mobility in radio context. Indeed, considering simulation in cases of cooperative networks, the number of cooperative agents in a given area is directly a parameter on the achievable positioning accuracy [11]. Another important aspect for producing a correct simulation is the habits of the agents. Indeed, Human motion are led by habits which can be directly translated into the mobility model

of the agents. Habits can be modeled e.g. from the propensity of an agent to be in a specific room of a building or to be under the influence of another agent.

Finally, a satisfying agent mobility description would be site-specific and include smart interactions of the agents both with their environment and with the other agents. A convenient solution to achieve a such deterministic mobility description is to use Discrete Event Simulation (DES). The mobility is intrinsically a space-time process in interaction with the environment. Practically, the simulation of such a phenomena can be quite complicated to address because it involves random variables (e.g. possible positions into the layout, transitions probabilities), which have to be dynamically modified by external events (e.g. other agents arrival). Those particular kind of problems can be advantageously managed by DES techniques which are designed for both stochastic and dynamic problems allowing the discrete evolution of variables [12]. The choice of DES has already been made by other mobility simulators oriented to urban mobility [13] because it allows the simulation of active components [14].

Another interesting aspect of using DES is the possibility to define autonomous entities. Practically it means that all the decisions of an agent is based only on its own observation of the environment. That autonomy can be limited by rules defined by a user (e.g. avoiding certain rooms of the environment). That precise implementation has been originally created for video games and the simulation of artificial intelligence. One implementation of those type of solution is based on the *steering behaviors* principles [15]. Steering behaviors allow the computer description of human behavior e.g. "follow a leader", "detect and avoid" or "path following". Based on these principles, a library called Personal Rapid Transit Simulation [16] has been enhanced to fit with the indoor localization applications.

B. The Network Simulation in Mobility

Although it exists commercial tools specialized in the pedestrian mobility to improve passenger flow efficiency in existing buildings or evacuation and traffic management [17], [18], [19], there is no such dedicated tools in the wireless communication field. Moreover, in commercial propagation tools (i.g. NS2, WSNET) the mobility is generally not taken into account. That lack could also be mentioned considering upper layers of the OSI standard, where the mobility parameters are rarely part of simulators. Although most of commercial tools take into account the mobility, it exists some research tools which integrate it into the network performances evaluation. For that purpose, and considering indoor situation, two models are generally considered:

- Cellular automata models which are based on geographic constrained model [20]: This method consists in the discretization the agent position on a grid. At each time step, each agent can move to a finite number of possible positions on that grid. Due to the ease of the method, the use of a grid for modeling the movement of the agent is often considered as a mobility model. In order to be more site-specific, some positions on the grid can

be forbidden, preventing agents to stand behind an open door for instance.

- Agent based models which are either spatio-temporal dependent or DES-based models: A more sophisticated approach which intends to reproduce a real pedestrian activity constrained by the environment. In that case, the use of DES based mobility model allows to retrieve in simulation phenomena which cannot be detected with a simple random walk or cellular automata model. Despite of their potentialities, only few simulators use that method in the considered context of radio simulations. Moreover, the question of the communication for distributed computation including the mobility seems to be lacking.

III. SIMULATION OF AGENTS MOBILITY

The proposed simulation of agents mobility is based on the *SimPy* framework which is a DES tool. The movement of agents is organized in two modes: the large scale mobility which represents the *motivation* of the agent (choosing a destination room) and the small scale mobility which constraints the agent to move and interact with the environment and the other agents.

A. Large Scale Mobility : A Graph Aided Description

At large scale, the agents mobility can be seen as a stochastic succession of two different states: Static State (SS) where the agent is static and Mobile State (MS) where the agent is mobile. Most of the time in indoor environment, mobile agents are static. Thus the static state SS can be fully considered as part of the overall mobility description. The large scale mobility consists in both defining a target for the agent and determining a path to reach that target.

1) *The Graph of Room \mathcal{G}_r and the Graph of Way Path \mathcal{G}_w :*
In order to determine the different targets of the agents, the graph of room \mathcal{G}_r is introduced. An example is represented in Figure 1. Designed especially to manage mobility, the graph of room is a simplified version of topological graph \mathcal{G}_t described in [1]. The nodes of \mathcal{G}_r represent rooms. For this purpose, the nodes of \mathcal{G}_r are the nodes of \mathcal{G}_t limited to cycles (\mathcal{G}_t 's nodes) with a door. In addition, each edge of \mathcal{G}_r interconnects rooms sharing a door. With such a description, each node can determine a target by choosing a room from the nodes of \mathcal{G}_r . Both the target and the path are determined with the help of graphs.

To obtain more realistic movement, the graph of way path \mathcal{G}_w representing all the possible transitions between rooms is introduced. The graph \mathcal{G}_w , represented in Figure 2 for the same environment, has the same "room" nodes of \mathcal{G}_r and extra nodes associated to different doors. Those nodes are added to encourage the agent both to cross the doors and to avoid unreal trajectory (e.g. going to the center of a corridor instead of directly reaching the opposite room). Nodes of \mathcal{G}_w are connected using two rules:

- All room nodes are connected to a door node
- The doors of two rooms are connected together if the 2 rooms are connected in \mathcal{G}_r .

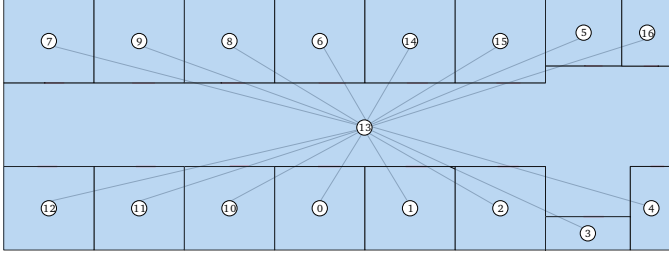


Fig. 1: An example of Graph of rooms \mathcal{G}_r

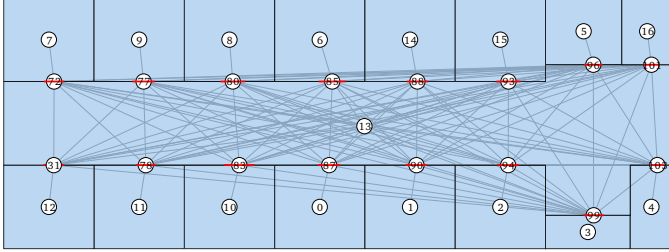


Fig. 2: An example of Graph of way path \mathcal{G}_w .

2) *Determination of a Target Using the Graphs:* The graph of room \mathcal{G}_r allows the agent to randomly choose a target (a.k.a a destination room) and the graph of way path \mathcal{G}_w allows to find a path to this target. That path is a succession of \mathcal{G}_w nodes. Let's suppose a situation where an agent in room i chooses a *targeted* room t with the help of graph \mathcal{G}_r . Rooms i and t correspond respectively to nodes v_w^i and v_w^t of \mathcal{G}_r . Then, the path from the room i to the room t is obtained by processing a Dijkstra [21] shortest path algorithm \mathcal{D} on nodes v_w^i and v_w^t of the graph \mathcal{G}_w :

$$\mathcal{V}_p = \mathcal{D}(v_w^i, v_w^t). \quad (1)$$

The obtained ordered set of nodes \mathcal{V}_p describes the shortest path between v_w^i and v_w^t . To achieve the whole trajectory from i to t , the agent has to successively go through all the rooms and/or doors corresponding to the nodes of \mathcal{V}_p . Each time a node of \mathcal{V}_p is reached, the next one becomes an intermediate target it , until the node v_w^t is reached. In Figure 2, an agent in room $i = 12$ with a target in room $t = 15$, corresponds to returned path $\mathcal{V}_p = \{12, 31, 93, 15\}$. Thus, the agent first moves from room 12 to its intermediate target 31 and so on until it moves to its final target in room 7.

B. Small Scale Mobility : Steering Forces

Once their targets are known and their paths have been determined, agents can start moving in the environment while avoiding walls and going through doors. This particular ability is defined as the small scale description of the mobility and is described with the help of steering behaviors. Steering behaviors have been introduced in [15] to describe interactions of agents with their environments. Agents are simply described by their mass m and two limiting parameters: their maximum acceleration a_{max} and their maximum velocity v_{max} . This description allows the agents to be driven by several steering forces applied on their center of mass. The computation of acceleration, velocity and position of agents is ensured by Euler integration [22]. At each time step the agent's acceleration

\mathbf{a} is re-evaluated in regards with the resulting steering force applied on it. The true steering forces values are truncated in regard of the maximum acceleration parameter of the agent a_{max} . The steering force \mathbf{F} applied on the agent is defined by:

$$\mathbf{F} := \max(\|\mathbf{F}\|, ma_{max})\hat{\mathbf{F}} \quad (2)$$

where $\hat{\mathbf{F}}$ is the unitary vector obtained from normalization of \mathbf{F} , and m is the mass of the agent. Then, the applied acceleration vector \mathbf{a}_n at time step n is given by :

$$\mathbf{a}_n = \frac{\mathbf{F}}{m} \quad (3)$$

As well, the true velocity \mathbf{v}_n at time step n is approximated by the Euler integration as the sum of the old velocity and the product of the current acceleration vector with $\delta\tau$, the time between $n - 1$ and n :

$$\mathbf{v}_n = \mathbf{v}_{n-1} + \mathbf{a}_n\delta\tau \quad (4)$$

Then, the applied velocity \mathbf{v}_n is truncated by the maximum velocity parameter of the agent v_{max} :

$$\mathbf{v}_n := \max(\|\mathbf{v}_n\|, v_{max})\hat{\mathbf{v}}_n \quad (5)$$

where $\hat{\mathbf{v}}_n$ is the normalization of \mathbf{v}_n .

The position \mathbf{p}_n at time step n is obtained by the Euler integration as the sum of the previous position \mathbf{p}_{n-1} and the current applied velocity:

$$\mathbf{p}_n = \mathbf{p}_{n-1} + \mathbf{v}_n\delta\tau \quad (6)$$

The steering forces applied on the agent correspond to human-like behaviors. In the following, only three types of steering forces are considered: the seek behavior, the obstacle avoidance behavior and the agent avoidance behavior.

1) *Seek Behavior:* The seek behaviors aims to produce a steering force which attract the agent to a target. In the considered scenario, those targets are succession of intermediate targets as defined in section III-A. The steering force direction will depends on the distance vector \mathbf{d}_n between the target \mathbf{t}_n and the actual position of the agent \mathbf{p}_n . Then, it is possible to determine a velocity vector \mathbf{v}_n^d defined with :

$$\mathbf{d}_n = \mathbf{t}_n - \mathbf{p}_n \quad (7)$$

$$\mathbf{v}_n^d = \max\left(\frac{\|\mathbf{d}_n\|}{\delta\tau}, v_{max}\right)\hat{\mathbf{d}}_n \quad (8)$$

where \mathbf{v}_n and \mathbf{v}_n^d are the current velocity vector and the desired velocity vector directed respectively; and $\hat{\mathbf{d}}_n$ is the normalization of \mathbf{d}_n . Thus, the velocity difference \mathbf{w}_n between the actual velocity \mathbf{v}_n and the desired velocity \mathbf{v}_n^d allows to compute the seeking force \mathbf{F}_s to be applied on the agent.

$$\mathbf{w}_n = \mathbf{v}_n^d - \mathbf{v}_n \quad (9)$$

$$\mathbf{F}_s = m \frac{\mathbf{w}_n}{\delta\tau} \hat{\mathbf{w}}_n \quad (10)$$

where $\hat{\mathbf{w}}_n$ is the normalization of \mathbf{w}_n and m the mass of the agent. This situation is represented in Figure 3.

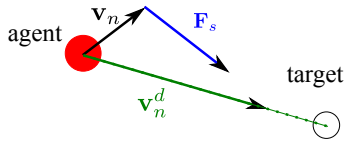


Fig. 3: Seek behavior: The steering \mathbf{F}_s vector results from the current velocity vector of the agent \mathbf{v}_n and the desired velocity vector \mathbf{v}_n^d directed to the target.

2) *Obstacle Avoidance Behavior:* The obstacle avoidance behavior aims to produce a steering force avoiding the agent penetrating a wall or any other element of the layout. The associated steering force \mathbf{F}_o is given by:

$$\mathbf{F}_o(\mathbf{d}_{k,w}) = \frac{m}{\delta\tau} \left((\mathbf{v}_{n-1} - \mathbf{v}_n) \frac{\alpha}{|\mathbf{d}_{k,w}|} \hat{\mathbf{n}} \right), \quad (11)$$

with $\mathbf{d}_{k,w}$ the distance vector from the agent k to a wall w , α a parameter properly chosen for avoiding the agent grazing the wall and $\hat{\mathbf{n}}$ a normalized vector orthogonal to the obstacle.

3) *Agent avoidance:* In [23], it is stated that it exists an inter-persons distance boundary. This boundary can be modeled with a circle with a radius r around the agent k . While another agent l penetrates inside that circle, an avoidance force F_a proportional to the distance vector between the agents $\mathbf{d}_{k,l}$ is applied. That force can be modeled using the previously defined obstacle avoidance force, regarding the 2 modifications:

- 1) The distance between the agent and the obstacle is replaced by the distance between the two agents,
- 2) A rotation matrix \mathbf{H} has to be introduced in order that the force be applied orthogonally to the velocity vector, and thus creating an avoidance of both agents.

Thus, the agent avoidance force F_a can be written:

$$\mathbf{F}_a(\mathbf{d}_{k,l}) = \begin{cases} \mathbf{H}\mathbf{F}_o(\mathbf{d}_{k,l}) & , \|\mathbf{d}_{k,l}\| \leq r \\ \vec{0} & , \|\mathbf{d}_{k,l}\| > r \end{cases} \quad (12)$$

with \mathbf{H} a rotation matrix:

$$\mathbf{H} = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \quad (13)$$

4) *Resulting Force:* An agent k is under the influence of different steering forces: a seek force which pull the agent to its intermediate target node v_{it} as seen in and several repulsive forces to avoid obstacles or other agents. The resulting steering force \mathbf{F} is written as the sum of all the steering forces applied on agent k :

$$\mathbf{F} = \mathbf{F}_s + \sum_w^W \mathbf{F}_o(\mathbf{d}_{k,w}) + \sum_{l \neq k}^K \mathbf{F}_a(\mathbf{d}_{k,l}), \quad (14)$$

where W is the number of walls in the vicinity of the agent k , which is determined with a simple distance threshold. The use of that threshold avoids the computation of forces which could be neglected due to the distance of the walls.

IV. APPLICATION TO LOCALIZATION AND NETWORK SIMULATION

The proposed mobility algorithm is developed to complement the graph-based propagation simulator PyLayers and to provide realistic simulations to evaluate localization and communication algorithms in dynamic and cooperative scenarios. This section proposes to illustrate some examples on how the dynamic simulation can be used with propagation tools and localization algorithms. A first scenario considers an agent moving into the environment while it is connected to 3 anchor nodes. That scenario is used to run the multi-wall algorithm to provide regularly location dependent parameters (LDPs) namely the *Time of Arrival* (ToA) and the *Received Signal Strength* (RSS) between the agent and the anchors nodes. Based on the same mobility simulation data, some CIR are computed using the RT propagation tool. Finally, a second simulation involving 2 mobile agents is proposed to demonstrate the communication stage.

A. Mobility Displaying

Figure 4 illustrates an example of agent mobility simulation. In this example, a single agent is moving into the layout and is in visibility of 3 anchor nodes. The three anchor nodes are represented with triangles. The different ground truth positions of the agent are sampled every second and plotted as red dots. Every 10 seconds the small dots are replaced by larger dots with an indexed variable t . Those larger dots are *checkpoint positions* and they allow to determine the direction of the agent and to give reference points to easily match the position and the LDP (i.e. RSS and TOA) values in the following figures.

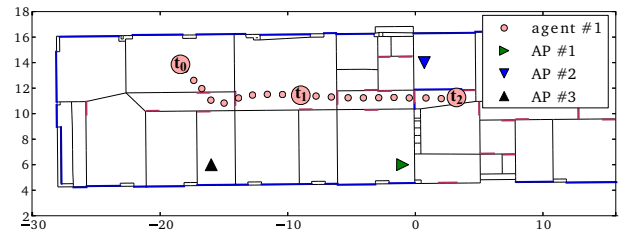


Fig. 4: Simulation of a trajectory of a mobile agent.

From the trajectory, it is possible to get the evolution of LDPs for the different radio links. Those LDPs can be obtained using Multi-wall algorithm or using the ray tracing tool [1]. Those LDPs are used as inputs for the localization algorithms. Figure 5, shows the evolution of both the received power and the delay introduced by the structure along the mobile agent trajectory with respect to the three anchor nodes.

B. Mobility for incremental ray tracing

One of the main goal of the PyLayers ray-tracing tool is to reduce the complexity and time consumption in ray determination using signatures. A ray signature is defined as both the succession of interactions (reflection, transmission, diffusion, etc) and structure segments (walls, doors,...) that a radio wave encounters from the transmitter to the receiver. When using graph description of the environment and the

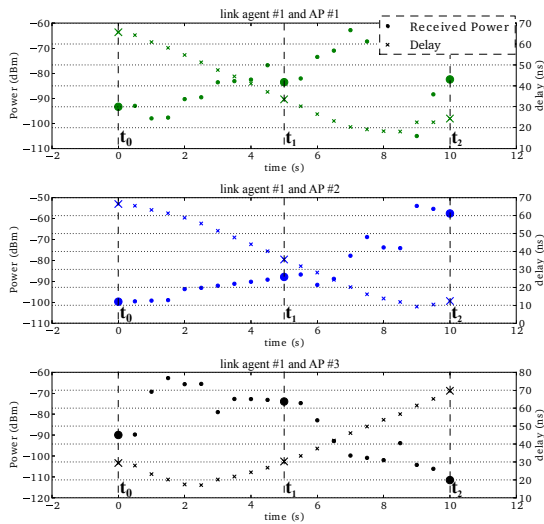


Fig. 5: The RSS (dots) and the estimated delay (cross) as a function of time for three different radio links.

interactions, signatures are obtained using a graph search algorithm. 2D and 3D rays are then easily obtained from signatures using geometric transformation [2]. Signatures have been introduced to represent the persistent component of the radio channel. It means that for a given environment, the channel perturbation are caused by some specific signatures which can be directly related to specific pattern into the channel impulse response (CIR).

Between 2 rooms, signatures are assumed to be stationary. Practically it means that while the transmitter or the receiver remain in their respective rooms, the same set of signatures can be used. It can be noticed that the signature description is only related to the environment and do not involve the position of neither the receiver nor the transmitter. Consequently, as long as an agent belong to a same room, the same set of signatures can be reused to calculate associated rays. As an illustration, CIR of the mobile agent is computed for points of its trajectory belonging to the same room. On the left column of Figure 7 are displayed the obtained rays between 6 positions of the agent and an access point. Facing these layout representations, the corresponding obtained CIR are displayed. The 6 set of rays and CIR have been obtained using the same set of signatures.

Figure 7 displays an histogram of the computation time for each CIR obtained in Figure 6. As expected, the computation of the first CIR is the longer due to signatures research step. The computation time required for obtaining the 5 other CIR is shorter because the same set of signatures is reused.

C. Example of Use of the Communication Stage

The simulation represented in Figure 8 is considered to demonstrate the communication stage between two agents. Two moving agents and four static anchors are considered. The two mobile agents start from different sides of the layout, move toward each other, cross themselves into a corridor and then continue to their destinations. The different trajectories of the agent 1 and the agent 2 are represented by red and

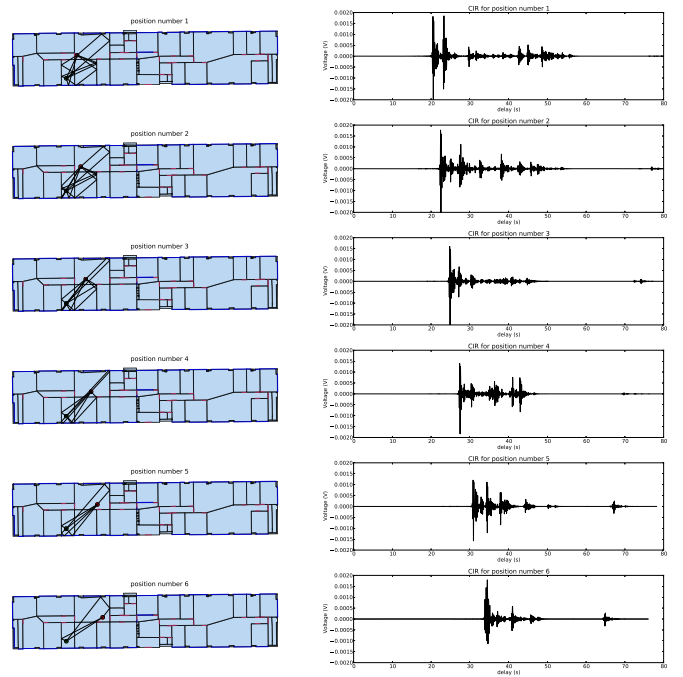


Fig. 6: Computed rays (left) and evaluated CIR (right) for different positions of the mobile agent along the trajectory.

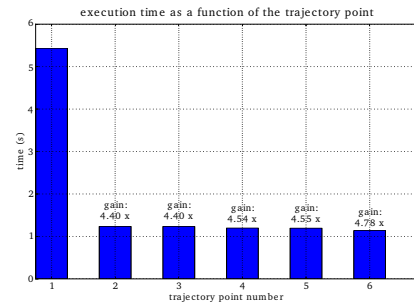


Fig. 7: Time required to produce CIR for 6 different positions of the mobile node. (Intel Core I7 with 4Gb of RAM)

white dots respectively. Let rat_1 , rat_2 and rat_3 three different radio access technologies (RAT) involved in the simulation. The mobile agent 1 embeds rat_1 and rat_3 and the mobile agent 2 embeds rat_2 and rat_3 . The two mobile agents can only be connected together on rat_3 . The mobile agent 1 is connected on rat_1 to the 2 static anchors represented with the red triangles, whereas the mobile agent 2 is connected on rat_2 to the 2 static anchors represented with the white triangles. Finally, each second, each agent tries to estimate its position using the available ToA. If not enough ToAs are available, the agent will try to communicate with its neighborhood. The communication stage has been configured to exchange both ToA and estimated positions between the two mobile agents.

Figure 9 represents the positioning error of the mobile agent as a function of the time. First of all, the zero values at the beginning correspond to an initialization value and mean that the agents have not yet start position estimation. On both curves, two regimes can be observed: A regime of large positioning error, corresponding to situations where mobile agents have only access to 2 ToA from the two anchor nodes;

and a regime of small positioning error where a third ToA is being available from the other agent.

Indeed, it is noticeable that a discontinuity appears in the positioning error at 5s for agent 1 and 4s for agent 2. Considering the checkpoint values, which corresponds to those represented in Figure 8, it appears that the decrease of the positioning error corresponds to a situation where the 2 mobile agents are in visibility. Then, the two agents communicate between each other in order to exchange both their estimated positions and a ToA value. As long as they are in visibility, the positioning error value remains low. At time 7 seconds, both agents loose their respective visibility and thus the possibility to exchange information. Then, the position estimation is no longer accurate and the positioning error increases again.

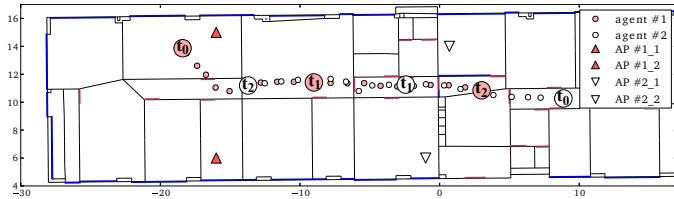


Fig. 8: The simulation scenario: two agents (red and green dots) and 4 anchor nodes (red and green triangles).

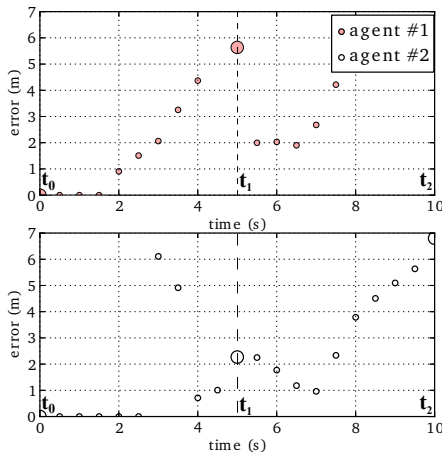


Fig. 9: Positioning error evolution. Between 4s and 7s the two agents are connected.

V. CONCLUSION

This paper presented a DES mobility algorithm designed for propagation and indoor localization simulation and developed as a part of PyLayers which is an open source wireless propagation and localization simulator. The algorithm is split in two types of mobility: large scale mobility based on the graph description and small scale mobility based on the steering behavior method. The algorithm has described how an agent can move into a given environment and interact with other agents. The main original part of that work is the adaptation of the steering behavior method to the graph representation. Some examples of simulations are presented to show the advantages of the proposed mobility algorithm especially in obtaining realistic human movements of agents and simplify ray tracing tools.

ACKNOWLEDGMENT

This work was performed in the framework of the *WHERE2* (ICT-248894) projects, which is partly funded by the European Commission and by the CORMORAN project (ANR11-INFR010), which is funded by the French National Research Agency (ANR).

REFERENCES

- [1] N. Amiot, M. Laaraiedh, and B. Uguen, "PyLayers: An open source dynamic simulator for indoor propagation and localization," in *Communications Workshops (ICC), 2013 IEEE International Conference on*, 2013, pp. 84–88.
- [2] M. Laaraiedh, N. Amiot, and B. Uguen, "Efficient ray tracing tool for uwb propagation and localization modeling," in *Antennas and Propagation (EuCAP), 2013 7th European Conference on*, 2013, pp. 2307–2311.
- [3] "PyLayers," <http://www.pylayers.org>.
- [4] N. Amiot, M. Laaraiedh, and B. Uguen, "Evaluation of a geometric positioning algorithm for hybrid wireless networks," *Software, Telecommunications and Computer Networks (SoftCOM), 2012 20th International Conference on*, pp. 1–5, sept. 2012.
- [5] M. Laaraiedh, L. Yu, S. Avrillon, and B. Uguen, "Comparison of Hybrid Localization Schemes using RSSI, TOA, and TDOA," *Wireless Conference 2011 - Sustainable Wireless Technologies (European Wireless), 11th European*, pp. 1–5, april 2011.
- [6] A. H. W. Hsu, F. Bai, *Mobility Modeling for Vehicular Networks*. Chapman & Hall CRC, 2012.
- [7] I. Rhee, M. Shin, S. Hong, K. Lee, S. J. Kim, and S. Chong, "On the levy-walk nature of human mobility," *IEEE/ACM Trans. Netw.*, vol. 19, no. 3, pp. 630–643, Jun. 2011.
- [8] J. Tian, J. Hähner, C. Becker, I. Stepanov, and K. Rothermel, "Graph-based mobility model for mobile ad hoc network simulation," in *Simulation Symposium, 2002. Proceedings. 35th Annual*, 2002, pp. 337–344.
- [9] F. Bai, N. Sadagopan, and A. Helmy, "IMPORTANT: A framework to systematically analyze the Impact of Mobility on Performance of Routing protocols for Adhoc NeTworks," in *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*, vol. 2. IEEE, 2003, pp. 825–835.
- [10] A. Jardosh, E. M. Belding-Royer, K. C. Almeroth, and S. Suri, "Towards realistic mobility models for mobile ad hoc networks," in *Proceedings of the 9th annual international conference on Mobile computing and networking*, ser. MobiCom '03. New York, NY, USA: ACM, 2003, pp. 217–229.
- [11] H. Wymeersch, J. Lien, and M. Win, "Cooperative Localization in Wireless Networks," *Proceedings of the IEEE*, vol. 97, no. 2, pp. 427–450, 2009.
- [12] L. M. Leemis and S. K. Park, *Discrete-event simulation: A first course*. Pearson Prentice Hall Upper Saddle River, NJ, 2006.
- [13] D. Krajzewicz, J. Erdmann, M. Behrisch, and L. Bieker, "Recent development and applications of SUMO - Simulation of Urban MOBility," *International Journal On Advances in Systems and Measurements*, vol. 5, no. 3&4, pp. 128–138, December 2012.
- [14] "SimPy Simulation Package," <http://simpy.sourceforge.net>, accessed: 2013-07-16.
- [15] C. W. Reynolds, "Steering behaviors for autonomous characters," in *Game Developers Conference*, vol. 1999, 1999, pp. 763–782.
- [16] "Personal Rapid Transit Simulation," <http://sourceforge.net/projects/prt/>, accessed: 2013-07-16.
- [17] "SimWalk: Pedestrian Simulation for Transport, Aviation and Architecture." <http://www.simwalk.com/>, accessed: 2013-06-13.
- [18] "PTV Viswalk," <http://vision-traffic.ptvgroup.com/en-us/products/ptv-viswalk/>, accessed: 2013-05-26.
- [19] "Legion Pedestrian Software," <http://www.legion.com/>, accessed: 2013-05-26.
- [20] V. J. Blue and J. L. Adler, "Cellular automata microsimulation for modeling bi-directional pedestrian walkways," *Transportation Research Part B: Methodological*, vol. 35, no. 3, pp. 293–312, March 2001.
- [21] E. Dijkstra, *A Short Introduction to the Art of Programming*. Holland, 1971.
- [22] L. Euler, *Introductio in analysin infinitorum*, ser. Introductio in analysin infinitorum. Apud Marcum-Michaellem Bousquet & Socios, 1748.
- [23] E. Hall, *The Hidden Dimension*. Doubleday, 1966.