

Layered Mobility Model Architecture - *LEMMA*

Alexander P. Pelov, and Thomas Noel

Abstract—This paper presents the generic layered architecture for mobility models (LEMMA), which can be used to construct a wide variety of mobility models, including the majority of models used in wireless network simulations. The fundamental components of the architecture are described and analyzed, in addition to its benefits. One of the core principles stipulates that each mobility model is divided in five distinct layers that communicate via interfaces. This allows their easy replacement and recombination, which we support by reviewing 19 layers that can form 480 different mobility models. Some of the advanced features provided by the architecture are also discussed, such as layer aggregation, and creation of hybrid and group mobility models. Finally, some of the numerous existing studies of the different layers are presented.

Index Terms—Framework, Mobility Models, Wireless Networks

I. INTRODUCTION

Nowadays, due to the exceptional success of wireless networks worldwide, it is difficult to imagine our lives without them – be it satellite communications or mobile phones. Yet, only a couple of decades ago they were far from being indispensable for the majority of us. During that time, a tremendous amount of efforts has been put in the improvement of these networks, which resulted in the creation

of a wide variety of protocols and technologies fulfilling many different requirements, ranging from Bluetooth and ZigBee, to UWB, Wi-Fi, WiMAX, GSM, and UMTS.

The major differences between wired and wireless networks are the shared communication medium and terminals' ability to change their positions. When addressing the latter issue, the majority of wireless technologies aim at maintaining any ongoing communications as long as possible by defining mechanisms to manage node mobility. Multiple such solutions have been proposed in regard to the various environments and particularities - e.g. routing packets in a centralized network such as UMTS is very different from the same task in a MANET. All these propositions were first tested in simulations in order to compare their performances and select the best solution, thus avoiding the development of full-featured implementations of all candidates. Because of this, it is very important to be able to draw valid conclusions from the simulations, by insuring that each subsystem and the simulation as a whole are as realistic as possible.

In this paper we are going to be addressing the part of the simulation governing nodes' movements - the mobility model. We are going to study microscopic mobility models [1] used in wireless network simulations. Many models have been proposed generating various types of movement patterns (e.g. vehicle in a city, pedestrian in a shop). Some of them are purely synthetic, while others are based on diverse collections of data or have well-established theoretical grounds. With that multitude of possibilities, it is still astounding to discover that the most popular mobility model by far is the Random Waypoint [2]. One of the major reasons for this is that the synthetic models are much

Alexander P. Pelov and Thomas Noël are with LSIIIT (Image Sciences, Computer Sciences and Remote Sensing Laboratory) a mixed research unit of CNRS, France and University of Strasbourg, France. You can contact them at the following address: LSIIIT, Pôle API, Bd Sébastien Brant, P.O. Box 10413, 67412 Illkirch, France (phone: +33.3.90.24.45.87; fax: +33.3.90.24.44.55; e-mail: {pelov, noel}@unistra.fr).

simpler and fit equally well or equally bad all scenarios. Complex models, on the other hand, are difficult to compare to each other and are generally specialized in a limited set of use cases. This does not incite the non-specialist to get acquainted with all their peculiarities in order to decide to what extent they may be useful and what possible adjustments could be made. Furthermore, following the classical model presentations, it is impossible to easily recombine different parts of several mobility models, which means that the general users are limited to the set of predefined scenarios, which may or may not correspond to their needs.

II. EXAMPLES

As an illustration, we are going to present two mobility models used in multiple simulation scenarios. Even though some of the details have been omitted for the sake of clarity, it is difficult to see right away how can one combine the different traits in order to obtain new models, or even – to what extent do these models differ. Later, we are going to give the LEMMA representation of these two models.

First, we are going to introduce the Random Waypoint Mobility Model (RWP) [3]. All nodes are confined in a 2D rectangle area. Node movement is generated by first selecting a destination point, and then moving at a constant speed until reaching it. The speed is uniformly distributed in a predefined range $[V_{min}; V_{max}]$. Once the destination point is reached, the node pauses for a time period selected from the range $[P_{min}; P_{max}]$. Afterwards, the procedure is repeated, until the end of the simulation. The execution of this model is illustrated in Fig. 1.

The Manhattan Mobility Model [4] is also defined in a 2D rectangle area. Node movement is restricted to an idealized street map containing only vertical and horizontal streets, each street having two lanes (one for each of the directions). Upon reaching an intersection, a node decides probabilistically whether to continue moving on the same street, or to turn left or right. The speed changes randomly on each time slot and depends on the maximal node acceleration, and the speed of the node preceding it on the same lane of the street (Fig. 2).

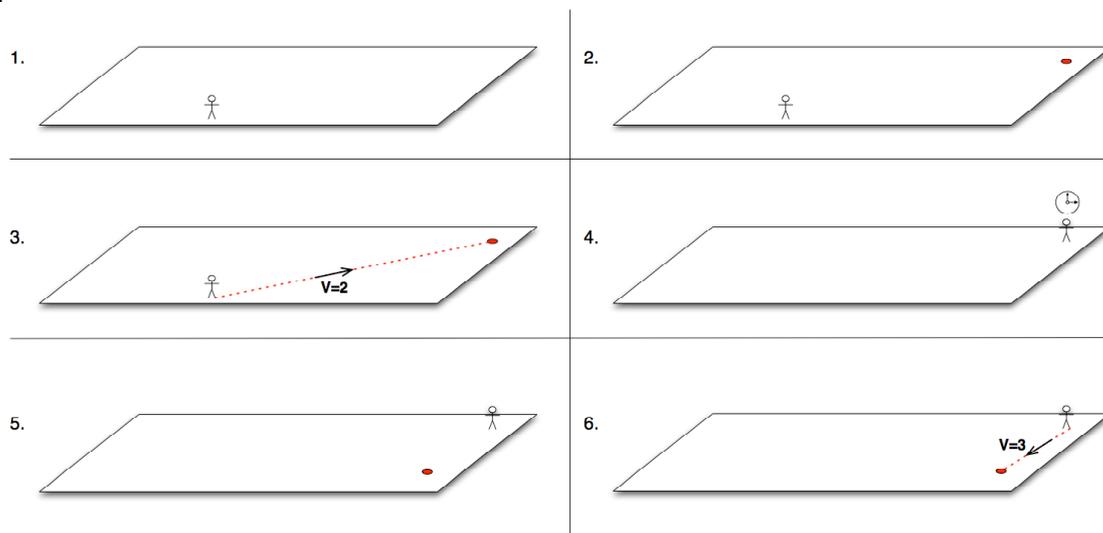


Fig. 1 - Execution of the Random Waypoint Mobility Model. First, the node selects the destination point, then moves constantly with a selected speed, and finally pauses. The process is repeated until the end of the simulation.

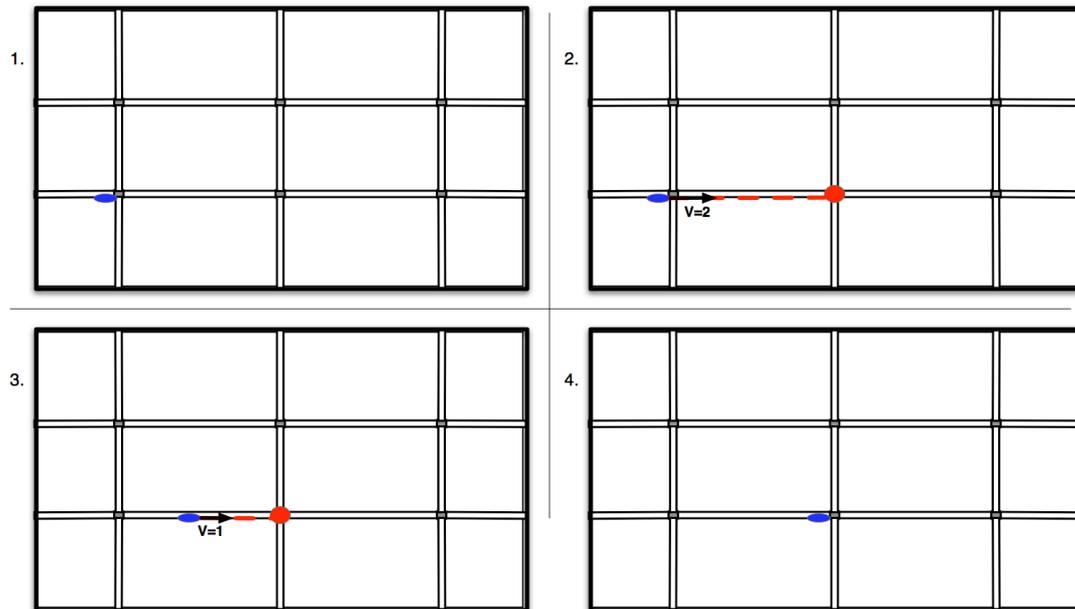


Fig. 2 - Execution of the Manhattan Mobility Model. First the node (given as a filled ellipse) probabilistically decides which direction it should take. Then, it moves by randomly changing its speed during its movement, until reaching the next crossroad, where the process repeats.

III. ARCHITECTURE DESCRIPTION

In this section we are going to describe in details the generic layered architecture for mobility models, first introduced in [44]. The architecture stipulates that a model is divided into several layers (Fig. 4), each layer having distinct, well-defined functions. Each layer exposes an interface, which can be used only by the layer directly above it, and its output is fed to the layer directly below it.

The usefulness and the viability of this approach have been proven in many existing systems, such as the TCP/IP family of protocols. Individual layers are less complex than the whole model itself, which helps simplify the development, the validation and the usage of new mobility models. The abstract description of the different layers and their interactions allows new propositions to be made and studied independently of the rest. Afterwards, they can be used in conjunction with any combination of existing layers. Several layers can be aggregated in order to fine-tune the behavior of the nodes, while sharing layer implementations across a set of nodes may simulate group behavior. Furthermore, this separation allows gaining a

better understanding of the influence of the different layers on the final result. Additionally, presenting an elaborate model in this form helps increase its readability, underline the major contributions, and insure that all necessary details are given, which is not always the case. Most importantly, because layers are functionally and semantically distinct, one can define specialized validation routines on a per-layer basis, e.g. by using study results and analyses performed in other research areas given later in this paper.

The movement of a node in a given environment may be regarded as a result of the interaction of a set of spatiotemporal processes. The environment is common for all nodes and contains all objects, properties and constraints, which may affect the movements of the nodes, such as points of interest and obstacles. The movement processes specify node's movements as a function of their environment and simulation parameters (Fig. 3). In the vast majority of cases, there is only one movement process, which governs the movement of all nodes in a simulation. However, one may find scenarios where each node has its own movement process (e.g. as in multi-agent

simulations), or processes that govern the movement of all nodes within a given subset of the simulation area, etc.

In order to prove the feasibility of LEMMA we have created a working implementation [42, 45].

A. Environment

The node environment is constituted of four types of entities, namely: simulation area, zones, constraints and movement influencing factors.

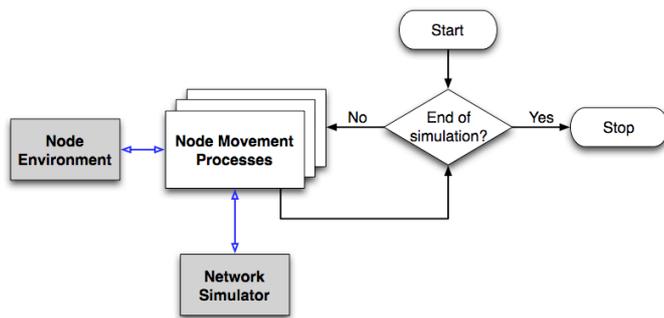


Fig. 3 - Movement process–node environment interaction.

• Simulation Area

The simulation area is the universe where the nodes “live” and move. It may be a one-, two- or three- dimensional space, or some other (user-defined) space. Each point P in this space is characterized by a set of coordinates (p_0, \dots, p_N) . The action to be taken if a node tries to move outside the simulation area depends on its border

behavior. In [5] Bettstetter summarized the bounce-back, wrap-around and delete-and-replace behaviors.

• Zones

The simulation area provides a sort of “low-level” coordinate-based positioning. However, people rarely think in terms of latitude and longitude - one would rather use names of places, cities, streets, buildings, etc. A zone is a connected set of points with an optional set of attributes. The zones form a high-level addressing space on top of the simulation area. They are a generic construct that can be used to represent a wide range of scenario entities, such as buildings, lakes, districts, desks, walls, etc. This allows the movement process layers to be defined in a scenario independent way. Moreover, the definition of zone operations is straightforward (union, intersection, ...), which enhances their expressiveness (e.g. picking all buildings from a given district).

• Constraints

Node movement is directly affected by the movement constraints, which restrict the possible movement trajectories. They depend on the movement process itself, e.g. a constraint may be in the form of a graph, a set of rectangles.

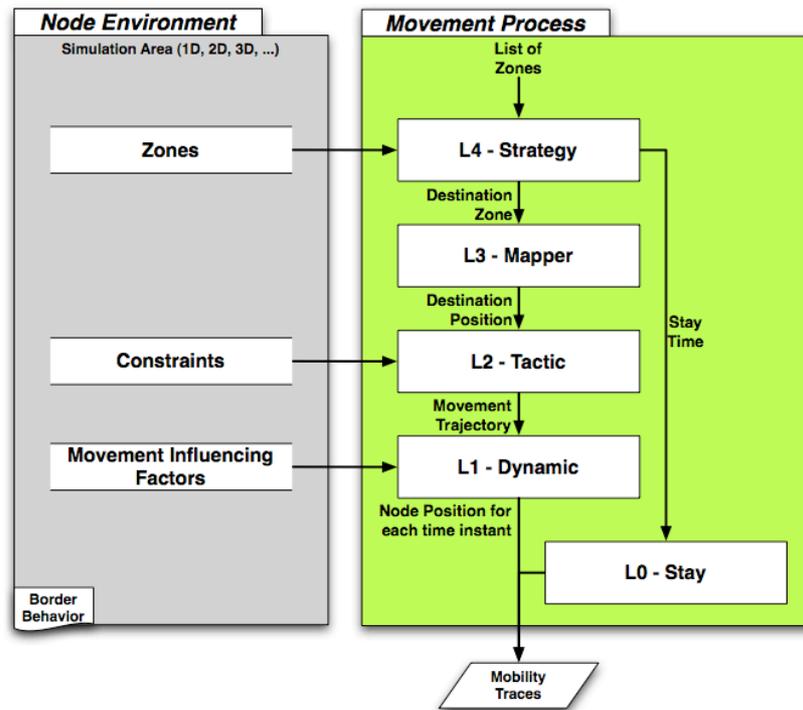


Fig. 4 - Node environment entities and movement process layers.

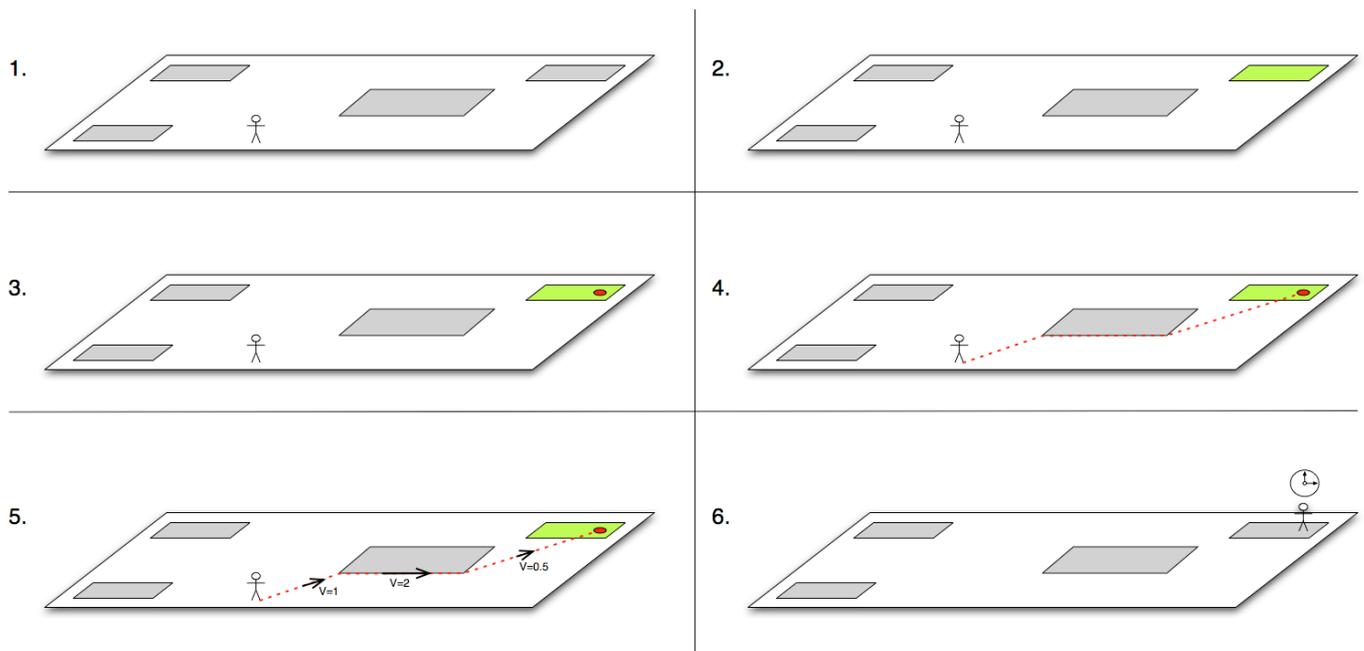


Fig. 5 - Example of a LEMMA execution. The strategy selects the destination zone in 2. The mapper chooses the exact position in 3, followed by the tactic, which selects the route to be followed in 4. In 5, the dynamic determines the speed at each point, and finally in 6, the stay layer specifies what should be done in this zone (e.g. pause).

• **Movement Influencing Factors**

Finally, the low-level aspects of the movement depend on the movement influencing factors. They are also dependent on the movement process and may include, amongst others, various traffic regulations (traffic lights, minimal/maximal speed, ...), rules that specify inter-node interactions (e.g. no collisions, speed matching), etc.

B. Movement Subprocesses

According to the layered architecture, a general node movement process can be divided into five layers called strategy, mapper, tactic, dynamics, and stay (as shown in Fig. 4).

A typical process execution has five stages (Fig. 5). First, the strategy selects the next zone to be visited. Then the mapper chooses a specific point from that zone, which allows the tactic to generate the trajectory. Finally, the stay layer determines the node movement during the stay period. The whole process is repeated until the end of the simulation, or until it is replaced (i.e. the node is assigned another mobility model). The purpose of the individual layers is detailed hereafter. There exist an abundance of studies on various mobility-related phenomena, which map directly to the different layers of our architecture. Some of them are also outlined in the following paragraphs.

• **Strategy**

The strategy layer represents the high-level movement decision-making process. It determines node's destination zones, but does not specify the movement trajectory itself. A strategy takes as input a list of zones it should choose from. The result of the execution of a strategy is the pair $\{next_zone, stay_time\}$, where *next_zone* is the next zone to be visited by the given node, and *stay_time* is the time it should stay in the selected zone after reaching it. Most often the nodes simply pause during the stay time, but one can specify a different movement process to govern their movement during the zone stay time.

Location prediction studies are strongly related to the strategy layer. Indeed, correctly

modeling the high-level movement patterns (determined by the strategy) is the goal of many of them. Extensive analyses have been performed in the area of cellular networks, and because of the direct correspondence cell \Leftrightarrow zone, the majority of them can be considered as direct implementations of the strategy layer. For example, Bhattacharya and Das [6] have proposed to create movement profile for each user, which is implemented as a Lempel-Ziv tree, whose alphabet holds one letter per zone. This idea is further developed and studied by Song et al. [7] and compared to Markov-based predictors.

Another class of relevant work includes the Origin-Destination studies performed by the traffic engineering researchers, e.g. [8]. Other methods used to create and/or validate strategy layer implementations include user activity modeling, and the related activity-based approaches, such as the ones discussed in [9], or Wi-Fi trace studies [43].

• **Mapper**

Having the next zone to be visited, the zone-to-coordinates mapper translates the high-level zone addresses generated by the strategy to "low-level" coordinates, which are then passed on to the tactic layer.

• **Tactic**

The tactic layer is the trajectory-generating process. It generates a route from point A (the current node position) to point B (the position supplied by the mapper), which satisfies the set of constraints set by the user.

Selecting routes between two given locations is also a subject of extensive studies, which can be regarded as studies of the tactic layer. The shortest path is not always the one being selected, as there may be other factors that affect people's decisions, e.g. traffic conditions, route attractiveness. It has even been shown that people may take different routes when doing a round trip [10].

• **Dynamic**

Finally, the movement dynamics layer specifies the speed and the acceleration, and possibly some small deviations from the trajectory that has been defined by the tactic.

Dynamics have also been thoroughly studied from many different points of view. If we concentrate on the case of vehicle movement, we will find multiple models of driver behavior, such as the velocity-difference model [11] or the Intelligent Driver Model [12]. Pedestrian movement requires other types of models, e.g. emergency evacuations [13].

- **Stay**

Upon reaching the end of its trajectory, and hence its destination zone, the node may be required to stay in it for a certain amount of time. During this time, the movement process defined in the stay layer governs its movement. In most of the models, the node simply pauses (i.e. it does not move), but there also exist models that define no zone stay time, the Random Waypoint (e.g. as in [14]), etc. The process set in this layer can be described with the same layered architecture, thus making any model described here an implementation of this layer.

IV. EXISTING LAYERS

Here are presented several examples of the environment components and mobility layers defined in the previous section. Most of them have been taken from existing mobility models, which helps illustrate the flexibility and the general character of the architecture.

A. Environment Components

- **Simulation Area**

In the huge majority of mobility scenarios, the simulation area is a 2D rectangle with a wrap-around or bounce-back border behavior (all studies using the Random Waypoint or the Random Walk mobility models fall in this category). Amongst the rarely used alternatives we can note the 3D parallelepiped area [15], and the fish bowl and Swiss flag [16].

- **Zones**

Zones with various shapes have been used in simulation scenarios - mainly rectangles, but also cubes and circles. These zones contain the points enclosed by the geometric figures, or their borders. We can also distinguish the singleton zone, which contains a single point from the simulation area.

- **Constraints**

The set of constraints used for mobility scenarios is limited. Almost all constraints can be expressed in terms of graphs embedded in the simulation area, zone avoidance or zone confinement. Indeed, most of the papers only describe the ways these constraints are configured, e.g. building graphs by using Voronoi paths [17], Delaunay triangulation [18], synthetic maps [4], real-world maps [19-23].

Almost all constraints can be expressed in terms of graphs embedded in the simulation area, zone avoidance or zone confinement. The embedded graph is a graph whose vertices are assigned a location (a point or a zone). A node obeying this constraint is restrained to move only on the edges and in the vertices of the graph. Zone avoidance restrains the conforming nodes from entering into zones marked to be avoided. This may be used to simulate walls, impenetrable buildings, etc. On the contrary, zone confinement keeps the nodes from leaving the zones marked to be their habitat. An example scenario may include rescue operation where different teams work in distinct areas of the simulation (as in [24]).

- **Movement influencing factors**

The factors influencing the movement dynamics vary, with the majority of cases using a simple constant speed movement. However many elaborated propositions exist, which change the speed of the nodes as a function of the speed or the number of the surrounding ones [20, 25-26], or add traffic signalization compliance [27-30].

B. Mobility Model Layers

A small sample of the layer implementations found in the literature is reviewed in the following paragraphs. The parameters for the strategies and the dynamics are given, while

for the tactics this is the case for the constraints they enforce. The presented mappers are parameterless. All these layers are given in a succinct, yet unabridged manner. In the typical use case, a scenario creator will pick one member from each layer and use it directly in his/her scenario.

• Strategies

1. Manually Specified

Zones - List of zones to visit, in the order to be visited.

Stay time - An ordered list of stay times.

The entire sequence of zones to be visited with the corresponding stay times is fixed in the scenario (e.g. as the movement of groups' logical centers in [31].

2. Uniform Random Zone

Zones - List of zones to choose from.

P_{min} , P_{max} - Minimum and maximum stay times.

The next zone to be visited is chosen from the list at random (with uniform distribution), and the stay in the zone is for a randomly selected amount of time (uniformly distributed in $[P_{min}, P_{max}]$) [3].

3. User-distributed Random Zone

Zones - List of zones to choose from.

$PD(z)$ - The distribution of the stay times, which depends on the zone.

$ZD(z,t)$ - The distribution from which the next zone is selected, as a function of time and zone.

The next zone to be visited is randomly chosen from the list, with a user-specified distribution $ZD(z,t)$, which depends on the current simulation time and zone. The stay time at the zone is drawn from the user-specified distribution $PD(z)$, which depends on the selected destination zone. This strategy is used in the WWP [32].

4. Mathematical Function

Zones - List of zones to choose from.

$F(t)$ - Vector function of the time t defining the motion.

For each moment of the simulation t , the smallest zone containing the point $F(t)$ is chosen and the stay time is fixed to 0, e.g. [33-34].

• Mappers

1. Fixed

The point to be selected is drawn from a list, which is specified by the creator of the scenario (e.g. [35]).

2. Random

Randomly select a point in the given zone, as in [14].

3. Random Border

Randomly select a point belonging to the border of the given zone, as in [36].

4. Gravity Center

Select the point of gravity of the given zone. It can be defined for an arbitrary zone as the sum of the radius vectors of all points in this zone, divided by the number of summands. Used in [37].

• Tactics

1. Linear

Does not enforce any constraints.

The node should move in a straight line from its current position to the destination point. Used in a wide variety of models, such as [14-15, 36-39].

2. Zone Avoiding - Shortest Route

Enforces zone avoidance constraints.

The movement is linear, with nodes never crossing zones marked as inaccessible. Instead, the shortest possible route is taken. If more than one shortest path exist, the one to be taken is randomly chosen.

3. Zone Avoiding - Border Route

Enforces zone avoidance constraints.

The movement is linear, with nodes never crossing zones marked as inaccessible. Instead, the node moves in straight line directly towards its endpoint until it reaches an inaccessible zone, which it surrounds tightly following the borders in a predefined direction (e.g. counterclockwise in 2D) until it is able to move on the straight line to its endpoint.

4. Zone Confining - Shortest and Border Route

Enforces zone confinement constraints.

Analogical to the Zone Avoiding cases, with the difference that nodes are allowed to move only on the zones marked as accessible.

5. Graph-constrained - Shortest Route Enforces embedded graph constraints.

Nodes obeying the constraints of an embedded graph can move only on its edges and in its vertices. That is, the movement is linear, following the edges of the graph. If a vertex is represented by a zone, the node takes the shortest path to the next edge. If the graph is oriented, then the movement follows the orientation of the edges. Finding a route in the graph is done by ignoring the size of the vertices and then selecting the shortest route (if the edges have associated weights, they are taken into account). This tactic is met in many of the map-based models (both synthetic and real-world based).

• Movement Dynamics

1. Constant Movement

V_{min} , V_{max} - Minimal and maximal velocity.

The node moves at a constant velocity, drawn uniformly from the interval $[V_{min}, V_{max}]$ for each trajectory. Very frequently used, as in [15, 32, 36, 38-39]. Setting $V_{min}=V_{max}$ will result in a frequently used scenario, where the nodes move at a constant speed throughout the simulation.

2. Smooth Random [1]

V - Set of velocities.

The speed selection is characterized by the use of a set of target speeds V (the speed a node intends to achieve) and a linear acceleration. The node moves with constant speed v until a random process draws a new target speed from the set of possible target velocities V . The node then accelerates (or decelerates) linearly until this desired speed is achieved (or a new target speed is chosen in the mean time).

3. Edge-limited [40]

The speed of a node depends on the number of nodes on the same edge of the graph. If there are many nodes on a given edge, the speed of all of them is reduced, which may be used to simulate overcrowding/congestion with little complexity.

4. Acceleration-constant-deceleration

V_{min} , V_{max} - Minimal and maximal velocity.

The node accelerates for a given time until reaching a randomly drawn speed in $[V_{min}, V_{max}]$, continues moves at a constant speed, and finally decelerates before reaching its destination.

5. Random [4]

V_{min} , V_{max} - Minimal and maximal velocity.

a_{min} , a_{max} - Minimal and maximal acceleration. Node velocity changes on each time slot, with uniformly selected acceleration in $[a_{min}, a_{max}]$. The speed is always kept in the range $[V_{min}, V_{max}]$.

6. Preceding-node-limited [4]

D - Base dynamic.

SD - Safety distance.

The node is governed by the based dynamic D , given as parameter. If the node is moving on a graph, and there is another node on the same edge of the graph, and the distance between the two nodes is less than SD , the speed of the base dynamic is limited to be at most the same as the preceding node.

C. Examples

If we return to the examples given beforehand - the Random Waypoint Mobility Model (RWP) and the Manhattan Mobility Model (MMM) - we can demonstrate the way they can be formed with the help of LEMMA. For both scenarios we choose a 2D simulation area with wrap-around border behavior. For the RWP we choose a single zone with the size of the whole simulation area. For the MMM we create one zone per intersection, all zones being vertices in a planar graph, one edge per street lane. At this point the environment for the two scenarios is completely described.

To represent the RWP movement process we may use the Uniform Random Zone strategy, the Random mapper, the Linear tactic and the Constant Movement dynamic, with a simple pause as a stay layer. For the MMM, we need the User-distributed Random Zone strategy with Random Border mapper, Graph-constrained Shortest Route tactic and Preceding-node-limited dynamic based on the Random dynamic.

This representation not only allows us to see the essence of each model clearly, but also provides the possibility to create new models based on the existing ones by simply changing a given layer, layer parameter or environment setup. For example, by replacing the strategy of the MMM to Uniform Random Zone strategy we obtain the City Section Mobility Model [41].

V. STRATEGY AGGREGATION, HYBRID AND GROUP MOBILITY MODELS

With the given architecture and the specified layers it is possible to create a wide variety of mobility models. However, there is more potential lying in the chosen approach. Indeed, it is possible to create advanced mobility scenarios, which use aggregated layers, or produce group or hybrid motion. In this section, we are going to describe the ways to achieve these powerful features.

A. Combining multiple strategies

Defining relations between zones is straightforward, as they are sets of points. As a consequence, since strategies take a set of zones as input, and output a single zone and stay time, it is possible to combine two or more strategies into a single one with the help of some inter-layer “glue” modules, which we are going to call strategy adaptors. Aggregating strategies may be done with the help of two simple types - zone and time adaptors (as shown in Fig. 6).

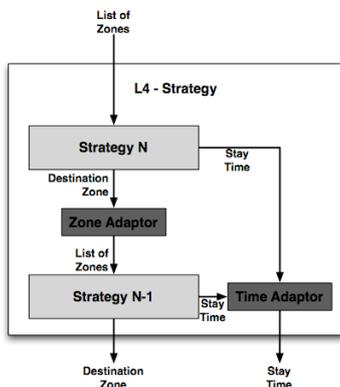


Fig. 6 - Using adaptors to combine strategies.

B. Constructing heterogeneous models

A node may be required to change its movement process during the course of the simulation, as for example pedestrians getting on and off busses, taxis, etc. These types of scenarios may be created by combining several mobility models, each representing a single kind of movement pattern. The movement of a node is then governed by a single “simple” model at a time, having a process selector to chose the active model according to some conditions, like current time, zone, surrounding nodes, etc. (see Fig. 7). The same principle is illustrated in Fig. 8 from the perspective of an individual node. The active mobility process for the node may be changed upon the occurrence of an event, and depending on the specified conditions. For example, one may use two movement processes – one to simulate daytime activities, and the other for the night movements, changing them every 12 hours.

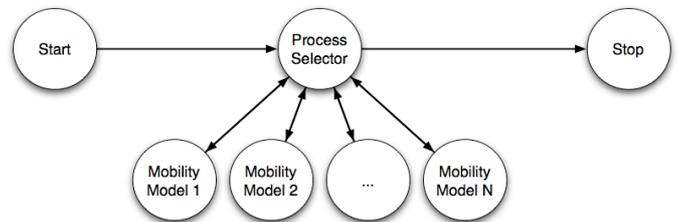


Fig. 7 - Constructing a new heterogeneous model by combining several mobility models.

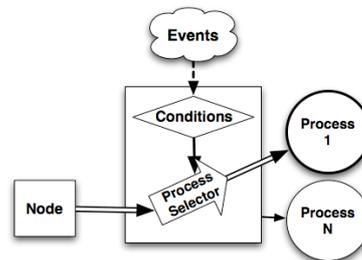


Fig. 8 - Process Selector schematic functioning - node perspective.

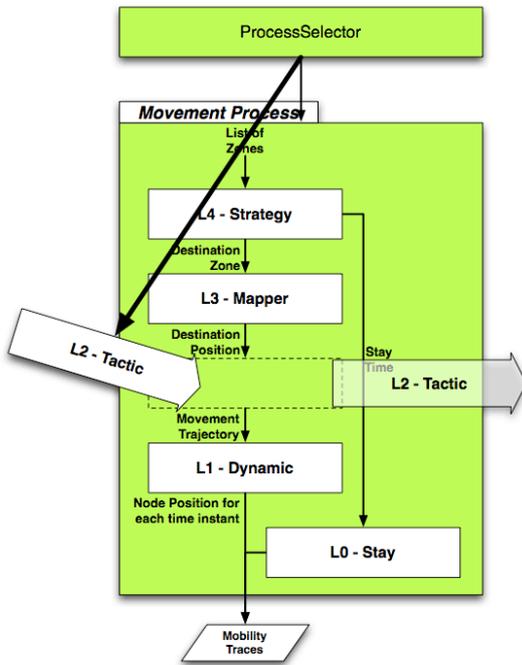


Fig. 9 - Fine-grained hybrid node behavior may be achieved by changing a single layer of its process.

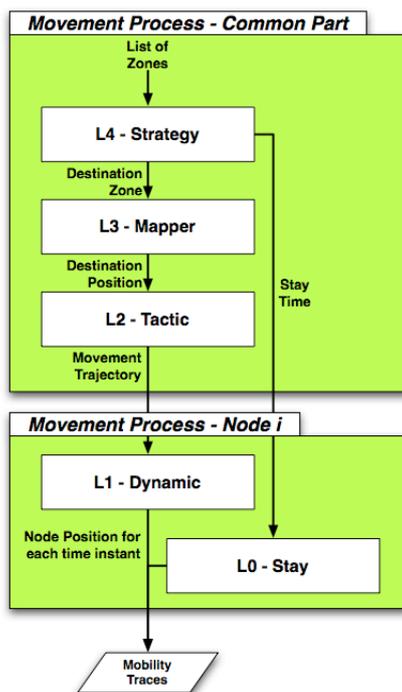


Fig. 10 - Several nodes may share the same layer instances. Here the nodes have common strategy, mapper and tactic layers.

Furthermore, it is possible to change a single layer during the model execution, which once again shows the flexibility of the selected approach. One can imagine a case, where the tactic is changed as a function of the destination point chosen by the mapper layer, as shown in Fig. 9. Using this

technique, it is easy to specify scenarios where nodes travel the long distances at high speeds, while handling the short distances at lower speeds (thus simulating getting on/off vehicular transport), and all this – without changing the individual layer definitions and implementations.

C. Creating group mobility models

Having defined the interfaces between all layers it is easy to share the instances of some of the high-level layers across several nodes, as shown in Fig. 10. This facilitates the creation of a group-like behavior, e.g. we may define the Reference Point Group Mobility Model (RPGM) [31] as having a shared strategy, mapper and tactic layers and using different parameters for the dynamic layer of each node (i.e. the specific reference point). Because, the RPGM is the most used group mobility model (most of the other models producing group movement patterns are its subset [41]), by following the example given in this paragraph, it is possible to obtain the majority of group-based movement traces used for wireless network research.

VI. CONCLUSION

The layered mobility model architecture was presented in this paper. It can be used to construct a wide variety of models, including the majority of microscopic models used in wireless network simulations. It divides the movement process in five layers, each having distinct, strictly defined functions, as well as the ways these layers interact. Additionally, the major entities of the simulation environment have been described, along with the way the different layers interact with them. The means of building heterogeneous and group models, and aggregating multiple layers have also been described.

REFERENCES

- [1] Bettstetter, C. (2001) Smooth is better than sharp: a random mobility model for simulation of wireless networks. , 19–27.
- [2] Kurkowski, S., Camp, T., & Colagrosso, M. (2005) MANET simulation studies: the incredibles. SIGMOBILE Mob. Comput. Commun. Rev. 9, 50–61.
- [3] Johnson, D. B. & Maltz, D. A. (1996) Dynamic Source Routing in Ad Hoc Wireless Networks. 353. .
- [4] Bai, F., Sadagopan, N., & Helmy, A. (2003) IMPORTANT: a framework to systematically analyze the Impact of Mobility on

- Performance of Routing Protocols for Adhoc Networks. 2, 825-835.
- [5] Bettstetter, C. (2001) Mobility modeling in wireless networks: categorization, smooth movement, and border effects. SIGMOBILE Mob. Comput. Commun. Rev. 5, 55-66.
- [6] Bhattacharya, A. & Das, S. K. (1999) LeZi-update: an information-theoretic approach to track mobile users in PCS networks. , 1-12.
- [7] Song, L. (2008) Evaluating Mobility Predictors in Wireless Networks for Improving Handoff and Opportunistic Routing.
- [8] Abrahamsson, T. (1998) Estimation of origin-destination matrices using traffic count--a literature survey.
- [9] Axhausen, K. W. & Gärling, T. (1992) Activity-based approaches to travel analysis: conceptual frameworks, models, and research problems. Transport Reviews 12, 323 - 341.
- [10] Golledge, R. (1995) Path selection and route preference in human navigation: A progress report. 988/1995, 207-222.
- [11] Jiang, R., Wu, Q., & Zhu, Z. (2001) Full velocity difference model for a car-following theory. Physical Review E 64.
- [12] Treiber, M., Hennecke, A., & Helbing, D. (2000) Congested Traffic States in Empirical Observations and Microscopic Simulations. Physical Review E 62, 1805.
- [13] Kuligowski, E. D. (2005) Review of 28 Egress Models. , 68-90.
- [14] Blazevic, L., Giordano, S., & Boudec, J. L. (2001) Self organized terminode routing simulation. 81-88.
- [15] Tudeuce, C. & Gross, T. (2005) A mobility model based on WLAN traces and its validation. 1, 664-674 vol. 1.
- [16] Le Boudec, J. Y. & Vojnovic, M. (2006) The Random Trip Model: Stability, Stationary Regime, and Perfect Simulation. Networking, IEEE/ACM Transactions on 14, 1153-1166.
- [17] Jardosh, A., Belding-Royer, E. M., Almeroth, K. C., & Suri, S. (2003) Towards realistic mobility models for mobile ad hoc networks. , 217-229.
- [18] Huang, D. (2005) Using Delaunay triangulation to construct obstacle detour mobility model. 3, 1644-1649 Vol. 3.
- [19] Nousiainen, S., Kordybach, K., & Kemppi, P. (2002) User Distribution and Mobility Model Framework for Cellular Network Simulations. , 518-522.
- [20] Saha, A. K. & Johnson, D. B. (2004) Modeling mobility for vehicular ad-hoc networks.. , 91-92.
- [21] Naumov, V., Baumann, R., & Gross, T. (2006) An evaluation of inter-vehicle ad hoc networks based on realistic vehicular traces. , 108-119.
- [22] Yoon, J., Noble, B. D., Liu, M., & Kim, M. (2006) Building realistic mobility models from coarse-grained traces. , 177-190.
- [23] Bratanov, P. (1999) User Mobility Modeling in Cellular Communications Networks.
- [24] Aschenbruck, N., Frank, M., Martini, P., & Tolle, J. (2004) Human mobility in MANET disaster area simulation - a realistic approach. , 668-675.
- [25] Tan, D. S., Zhou, S., Ho, J., Mehta, J. S., & Tanabe, H. (2002) Design and Evaluation of an Individually Simulated Mobility Model in Wireless Ad Hoc Networks.
- [26] Stepanov, I. (2002) Integrating Realistic Mobility Models In Mobile Ad Hoc Network Simulation.
- [27] Markoulidakis, J. G. (1997) Mobility Modeling in Third Generation Mobile Telecommunication Systems. IEEE PCS
- [28] Kim, J. & Bohacek, S. (2005) A Survey based Mobility Model of People for Urban Mesh Networks. MeshNets'05
- [29] Ilyas, M. U. & Radha, H. (2005) The influence mobility model: a novel hierarchical mobility modeling framework. 3, 1638-1643 Vol. 3.
- [30] Potnis, N. & Mahajan, A. (2006) Mobility models for vehicular ad hoc network simulations. , 746-747.
- [31] Hong, X., Gerla, M., Pei, G., & Chiang, C. (1999) A group mobility model for ad hoc wireless networks. , 53-60.
- [32] Hsu, W., Merchant, K., Shu, H., Hsu, C., & Helmy, A. (2005) Weighted waypoint mobility model and its impact on ad hoc networks. SIGMOBILE Mob. Comput. Commun. Rev. 9, 59-63.
- [33] Tolety, V. (1999) Load reduction in ad hoc networks using mobile servers.
- [34] Bergamo, M. & others (1996) System Design Specification for Mobile Multimedia Wireless Network MMWN.
- [35] Minder, D., Marrón, P. J., Lachenmann, A., & Rothermel, K. (2005) Experimental construction of a meeting model for smart office environments.
- [36] Hong, D. & Rappaport, S. S. (1986) Traffic model and performance analysis for cellular mobile radio telephone systems with prioritized and nonprioritized handoff procedures. Vehicular Technology, IEEE Transactions on 35, 77-92.
- [37] Scourias, J. & Kunz, T. (1999) Activity-based mobility modeling: realistic evaluation of location management schemes for cellular networks. Wireless Communications and Networking Conference, 1999. WCNC. 1999 IEEE 1
- [38] Basagni, S., Chlamtac, I., Syrotiuk, V. R., & Woodward, B. A. (1998) A distance routing effect algorithm for mobility (DREAM). , 76-84.
- [39] Johansson, P., Larsson, T., Hedman, N., Mielczarek, B., & Degermark, M. (1999) Scenario-based performance analysis of routing protocols for mobile ad-hoc networks. , 195-206.
- [40] Breyer, T., Klein, M., Obreiter, P., & König-Ries, B. (2004) Activity-based user modeling in service-oriented ad-hoc-networks.
- [41] Camp, T., Boleng, J., & Davies, V. (2002) A Survey of Mobility Models for Ad Hoc Network Research. Wireless Communications & Mobile Computing (WCMC): Special issue on Mobile Ad Hoc Networking: Research, Trends and Applications 2, 483-502.
- [42] Pelov, A. and Noel, T. (2009) Creating advanced mobility models with LEMMA. In Proceedings of the 12th Communications and Networking Simulation Symposium (San Diego, CA, USA, March 22-27, 2009). CNS '09. ACM, New York, NY
- [43] Pelov, A.; David, P.; Noel, T., "Trace Analysis of a Wireless University Network with Authentication," Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks and Workshops, 2007. WiOpt 2007. 5th International Symposium on , vol., no., pp.1-6, 16-20 April 2007 URL: <http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=4480111&isnumber=4480002>
- [44] Pelov, A.P.; Noel, T., "Layered Architecture for Mobility Models - LEMMA," Broadband Communications, Information Technology & Biomedical Applications, 2008 Third International Conference on , vol., no., pp.365-372, 23-26 Nov. 2008 URL: <http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=4696134&isnumber=4696070>
- [45] URL: <http://www.mobilitymodels.org/lemma>



Alexander P. Pelov is PhD Candidate at LSIIIT, Illkirch, France – a mixed research unit between CNRS, France and University of Strasbourg, France. He has finished his Masters degree in Computer Science at University of Provence, France. His research interests include wireless network mobility models, and Mobile and Vehicular Ad-Hoc Networks. He enjoys playing chess, reading and going to the cinema.



Thomas Noel is professor at University of Strasbourg, France. His research activities include several aspects of wireless communication networks and telecommunication systems. He is particularly interested in self-organized mobile networks, mobile network architecture and protocols, ubiquitous computing and multicast and group communications.