PH.D. THESIS

PRESENTED AT:
UNIVERSITY OF STRASBOURG
DEPARTMENT OF MATHEMATICS AND COMPUTER SCIENCE
LSIIT (CNRS UMR 7005)

FOR OBTAINING THE DEGREE:
UNIVERSITY OF STRASBOURG
DOCTOR OF PHILOSOPHY (PH.D)
IN COMPUTER SCIENCE

MOBILITY MODELS FOR WIRELESS NETWORKS

by

ALEXANDER PELOV

PUBLIC DEFENSE ON DECEMBER 15TH, 2009 WITH THE FOLLOWING JURY:
FABRICE VALOIS, *External evaluator, Professor at INSA Lyon*
MARCELO DIAS DE AMORIM, *External evaluator, CNRS Research Scientist*
THOMAS NOËL, *Thesis advisor, Professor at University of Strasbourg*
JEAN-JACQUES PANSIOT, *Examinator, Professor at University of Strasbourg*
CHRISTIAN BONNET, *Examinator, Professor at EURECOM Institute*

To my family.

Dedicated to the loving memory of
Blagorodna Bozhinova /1925 – 2000/
Pelo Pelov /1919 – 1997/ and
Zdravko Petrov /1947 – 2010/

ABSTRACT

Wireless networks have witnessed an explosive development in the past few decades, both for civil and military uses. The wide variety of requirements and application scenarios have provided an abundance of research challenges, some of which encountered for the first time in the context of computer network communications at such significant scale.

One of the major reasons of the success of these networks is the possibility to remain mobile while still using some of the services provided by the network. As a consequence, mobility modeling has become a first class actor of wireless network related studies.

In this thesis we have studied the various aspects and properties making a mobility model appropriate for wireless network research, including but not limited to analytical and simulation studies of WLAN, MANET, VANET, DTN and Cellular networks.

We have introduced the Layered Mobility Model Architecture (LEMMA), a general framework which facilitates the creation, modification, validation and verification of mobility models. The architecture is based on three simple principles enforcing few restrictions on the model components, and in the same time providing great flexibility. We have formulated the foundations necessary for defining and studying analytical models by following the principles of this architecture. Additionally, we have proved essential mathematical properties of the framework and have demonstrated both empirically and formally that any mobility model can be represented with LEMMA.

In order to provide a strict correspondence to the intuitive idea of model realism we have formalized the validity of a model in a given context. Finally, we analyzed two real-world GPS trace data sets with and defined a context of validity based on these traces.

RESUME

Depuis quelques années, les réseaux sans fil connaissent un véritable engouement aussi bien dans le domaine civil que militaire. Les spécificités de ce type de réseaux ainsi que la grande variété de scénarios d'application ont fait émerger de nouveaux défis dans la recherche, notamment dans le domaine des réseaux informatiques.

L'une des principales raisons du succès de ce type de réseau réside dans la possibilité de se déplacer tout en continuant à bénéficier des services offerts par le réseau. La modélisation des déplacements est par conséquent devenue l'une des thématiques de recherche de premier plan dans le domaine des réseaux sans fil.

Dans cette thèse, nous avons étudié les différents aspects et caractéristiques qui font qu'un modèle de mobilité est valide pour l'étude des réseaux WLAN, MANET, VANET, DTN et cellulaires.

Nos travaux nous ont amené à la proposition d'une nouvelle architecture appelée Layered Mobility Model Architecture (LEMMA). Cette architecture facilite la création, la modification, la validation et la vérification des modèles de mobilité. LEMMA est basé sur trois principes simples qui imposent très peu de restrictions sur les composants d'un modèle et qui en même temps offrent une grande flexibilité. En outre, nous avons défini les fondations nécessaires pour définir et étudier des modèles de mobilité analytiques en suivant les principes de cette architecture. De plus, nous avons prouvé les propriétés mathématiques de notre architecture et nous avons démontré empiriquement et formellement que tout modèle de mobilité peut être représenté avec LEMMA.

Afin d'établir une correspondance avec l'intuition qu'un modèle est réaliste, nous avons formalisé la validité d'un modèle de mobilité dans un contexte donné. Enfin, nous avons analysé deux ensembles de traces GPS provenant de mesures sur le terrain pour la définition d'un contexte de validité basé sur ces traces.

ACKNOWLEDGMENTS

First and foremost, I would like to express my deepest gratitude to my advisor Thomas Noël. I have rarely met someone who has put so much trust and patience in me, and whose constant support and precious advices were always there when I needed them. Thomas, thank you!

I would like to thank the members of my jury Fabrice Valois, Marcelo Dias de Amorim and Christian Bonnet for accepting to review this work. Their advice and patience is very appreciated. I am also very grateful to have Jean-Jeacques Pansiot in my thesis committee. He is a remarkable researcher and has always been an inspiring interlocutor for me.

A special thanks goes to Julien Montavont and Vincent Lucas for the countless fruitful discussions, for always being on my side in the unequal battle with the French language and administrative work, and most of all for their friendship!

Some of the work presented in this thesis wouldn't happen without the precious involvement of some people I would like to mention here. Many thanks to Jean Lorchat and Koshiro Mitsuya whose invaluable advices concerning Nagoya's GPS traces were crucial for my progress. Special thanks to Sébastien Vincent for sharing his evenings in discussions and coding on NS3 and LEMMA projects. Additionally, I want to thank Sébastien Boggia and Christophe Saillard for the indispensable help with the university WLAN traces!

These years wouldn't be the same without my lab colleagues, who became my friends and who made it so much fun. Romain Kuntz, Antoine Gallais, Stephane Cateloin, Guillaume Schreiner, Julien Gossa, Martin Andre, Pascal Merindol thank you for being as you are and so much more than I've ever hoped! Special thanks to Emil Ivov who incited me to come to Strasbourg thus helping this adventure to happen.

I could not continue without the support of all friends I had around me - Vlado and Mimi, Shadi and Waed and their little baby Zoya, Ahmed and Yosra, Edwin and Christelle, Sara, Caroline Thomann, Boyan, Petia and Draga, Caroline Schelske and Itso, just to name a few. Thank you all!

I would like to express my gratitude to the teachers who have strongly influenced me during my school and university years and have always been a source of inspiration and a model for me - Zdravko Petrov, Blagovesta Goranova and Donka Nikolova. Your faith in me gave me the courage in many difficult moments, and your teachings helped me become the person I am today.

Last but not least, I would certainly not be here if it wasn't my family, who has always encouraged me to follow my heart and inquisitive mind in any direction that might take me and has always been there for me in the moments of doubt. Mom, Dad, Sara and Yoan thank you!

And after the last, and very far from the least, I would like to thank Yana from the bottom of my heart for sharing this whole adventure with me (even though normally she avoids extreme activities), for supporting me and for still loving me after all these years!

CONTENTS

ACRONYMS

AP Access Point

APs Access Points

BIC Bayesian Information Criterion

CD Contact Duration

CMM Clustered Mobility Model

DTN Delay/Disruption Tolerant Network

ESRI Economic and Social Research Institute

GDF Geographic Data Files

GIS Geographic Information System

GPS Global Positioning System

HRW Heterogenous Random Walk

ICT Inter-Contact Time

LEMMA Layered Mobility Model Architecture

LMR LEMMA Model Representation

MANET Mobile Ad-Hoc Network

MLE Maximum Likelihood Estimation

OD Origin-Destination

RD Random Direction

RMSE Root Mean Square Error

RPGM Reference Point Group Mobility

RW Random Walk

RWP Random Waypoint

SIMPS Sociological Interaction Mobility for Population Simulation

SLAW Self-similar Least-Action Walk

VANET Vehicular Ad-Hoc Network

WLAN Wireless LAN

WSN Wireless Sensor Network

1 INTRODUCTION

They say that every atom in my body
was once a part of a star.
A star that had to die
so that I could exist.
Here and now.
Wandering and wondering.
What is the purpose of all this?
Of all these neighbors of mine?
And all the obstacles I simply ignore?
And all that space that lies before me?

Just waiting to be conquered.

How long have I been here?
At that very spot.
Frozen.
Motionless.
Am I really free?
Or am I just a pawn
in the game of the destiny?
Where should I be going next?
So many options and so little time!

Does it matter at all
if I choose my destination based on reason?
Or is it better just to throw a coin?
And have faith.
And never turn back.
Pursue the goal relentlessly,
until I gloriously reach it!
And stop to listen to the silence of success.

Oh, god, did that star really had to die?

The Node and the Random Waypoint Mobility Model

The importance of the computer networks has grown tremendously in the past couple of decades, and even more in the recent years. It has come to the point where the Internet access is considered a human right in some countries [Wik09a]. Accessing wirelessly the ever increasing number of services available on the Internet has become an integral part of its rapid development. Additionally, ad-hoc networks present many opportunities to establish communication whenever there is no infrastructure. These solutions pose a whole new set of problems to be resolved, such as dynamic topology change, varying connection quality and intermittent connectivity which are all a direct consequence of terminals' mobility.

The need of studying the effects of mobility and finding ways of addressing the emerging issues is evident in the light of the aforementioned facts. Modeling the mobility of the terminals therefore is of primordial importance to understanding and designing solutions adapted to the current and future developments of wireless networks. This work is aimed at understanding the needs of mobility model researchers - both model users and creators, and providing answers to their problems.

1.1 BACKGROUND

The development of the mobility models through the years has followed the availability of computer power and the types of networks being studied. Initially most of the works were predominantly analytical studies which used only simple, mathematically tractable models, such as performing random walk on a 2D lattice. Gradually, with the increased solution complexity and the proliferation of wireless network applications, the focus shifted to simulation-based studies, which opened the way to more complex mobility models.

Even though it is difficult to define general model evaluation criteria, increased complexity is not by itself a measure of the goodness of fit of a model, be it universally or for a specific context. Having more parameters might allow finer control over the modeled mobility patterns, but this is not always beneficial even if all parameters are meaningful and easy to select. This is one of the reason why traffic and pedestrian microsimulations have not found widespread acceptance in the wireless network research studies, even though they are considered to be able to realistically model movement patterns.

Traffic and pedestrian microsimulations generate the movement patterns of each individual (vehicle or pedestrian) by taking into account its own behavior and the patterns of all surrounding individuals. The process is time consuming not only because the decisions of individuals must be reevaluated at every time step, but mostly because achieving realistic behavior requires the simulation of an entire population of possibly unrelated individuals, e.g. obtaining the movements of a single bus still demands the simulation of entire region of the city. The biggest obstacle before the acceptance of such detailed mobility pattern generation methods is related to the significant efforts needed to correctly setup the microsimulation parameters. Indeed, the purpose of these models is to evaluate traffic conditions (such as traffic jams, lane merging, etc.) or pedestrian movements in a specific scenario (e.g. building evacuation), which are typically part of much bigger, policy-making projects. Consequently, the configuration of such microsimulation is typically done by a team of specialists, which have at their disposition abundant amount of relevant data, such as population density, income, occupation, family status, etc. and who validate the scenario before accepting the chosen parameters as appropriate for the given study case. Even if all these points are fulfilled, the results vary widely depending on the used microsimulator, and are not guaranteed to be realistic [CMS03]. Finally, some network simulation scenarios of special interest are out of the scope of traffic and pedestrian microsimulations, e.g. Mobile Ad-Hoc Network (MANET).

1.2 PROBLEM STATEMENT

Wireless network studies can be roughly divided into three categories with respect to the mobility they use - *no mobility/not applicable*, *defining* *new model* or *using existing model*. The papers defining own mobility model are much less than the ones using existing models [KCC05], which shows that mobility models are typically used as tools. This is an important aspect to be considered when designing and implementing such models. Indeed, in light of this aspect the ease of use and the simplified model understanding become factors of major importance - something often neglected in mobility model proposals. Therefore, a good mobility model should possess all of the following important aspects:

1. Be accessible and easy to use.
2. Be sufficiently detailed and easy to understand.
3. Have understandable parameters.
4. Have known mathematical properties.
5. Have a clearly defined scope of validity.

Throughout the chapters of this thesis we have aimed to address each of these points in a consistent way. We have created a platform, which provides the foundations for designing mobility models possessing the above criteria.

1.3 THESIS OUTLINE

This thesis is organized around these major properties that have to be present in order a model to be appropriate for simulation-based and analytical studies, and in the same time be attractive to a wider public of wireless network researchers. It is divided into three parts.

The first part (Part i) includes general and background information on the subject. It includes the definitions needed to formalize model realism as described in Chapter 2, and Chapter 3 which presents survey of the mobility modeling-related works found in the literature

The second part (Part ii) is focused on the introduction and the analysis of the Layered Mobility Model Architecture (LEMMA). Chapter 4 described the framework together with its core principles and the various resulting consequences and features. The mathematical properties of the architecture are then studied in Chapter 5, where are demonstrated some fundamental theorems regarding the universality of LEMMA.

The third part (Part iii) provides an example of the definition of context of validity based on two independent datasets of GPS traces. The context is based on taxi movements and can be used to measure the fitness of mobility models.

The final part (Part iv) presents the general conclusions of this work and the possible future research directions.

In the appendix, we have included several elements which are important to the arguments of the universality of the architecture, such as an empirical survey of the possible LEMMA representations of multiple existing mobility models (Section A.1), and the description the LEMMA simulator we have developed (Chapter B). Furthermore, we have included a survey of several of the existing mobility models (Sections A.2 and A.3) which have important aspects and/or popularity and as such cannot be omitted from our study. Finally, we have provided the long proofs of theorems and lemmas in Section A.4. Even though we consider that these works are important to the general character of the architecture and the completeness of its introduction, they are non-essential to the understanding of its principles and applications. Therefore we have preferred to improve the readability of this thesis and present only the essential parts of our approach, leaving the additional information to be consulted upon necessity.

Part I

GENERAL INFORMATION

2 MODEL REALISM

One of the most frequent claims used by mobility model designers is that a specific model is "realistic" or at least "more realistic" than another one (which in the majority of cases is the Random Waypoint (RWP)). There are many problems with these kinds of statements, the most important of which is that for the moment there is no widely accepted definition of *what* realistic mobility model is. In fact, very few of the authors give any definition at all (e.g. [MSK+05, SK99a, CB05, COdM04] refer to a model or a scenario being realistic without further investigating the question), and even when they do many of the choices remain unexplained or unsupported by evidence. In this chapter we are going to introduce our understanding of what should be the criteria of a "good" mobility model, what do we mean when we say that a model is "realistic", and how do the two concepts relate to each other.

2.1 VALIDATION AND VERIFICATION

Various kinds of models have been used in almost all branches of the science (e.g. economics [Marar]) and have been investigated as such from many perspectives. Even though the notion of realistic model is somewhat intuitive, it can be difficult to decide whether if a model is realistic or not based on informal perceptions, which are often too vague and may sometimes be misleading. Indeed, as [RBA05] demonstrates, it is possible to find real-world scenarios even for purely synthetic, stochastic models as the RWP. Thus instead of aiming at putting a label on some of the models based on arbitrary criteria, one should try to explicitly state the assumptions behind a model and its purpose, and perform a throughout *verification*[1] and *validation* [2].

Although an important part of the mobility models found in the literature were validated to some degree, most of them were not systematically studied, even though in [Sar98] the author points out that the validation process is most commonly carried out from its development team. In [Car96] the author suggests that validation should not be held up as a pre-requisite for the presentation of a computational model and its predictions, because of the delays and difficulties related to validating and presenting a model in a timely and succinct manner, but the problem is that often the presented models are never analyzed further. Sometimes the explanation put forward is that there are no existing real-world testbeds and traces, but even in these cases the analysts can perform at least sensitivity analysis[3], even though in this case strong validation claims are impossible [Kle99].

1 "Ensuring that the computer program of the computerized model and its implementation are correct" as defined in [ea79].
2 "Substantiation that a computerized model within its domain of applicability possesses a satisfactory range of accuracy consistent with the intended application of the model" as defined in [ea79].
3 Sensitivity analysis, as defined in [Kle99], is the systematic investigation of the reaction of the simulation responses to extreme values of the model's input or to drastic changes in the model's structure.

A simplified version of general modeling process is shown in Fig. 2. It presents how model verification and validation relate to the model development process [Sar98]. The model is developed by defining the *problem entity* to be modeled, the *conceptual model* to be used to that end, and the way it is going to be programmed in a *computerized model*. In mobility modeling, the *problem entity* represents the microscopic movements of various real-world objects (pedestrians, vehicles, animals, etc.), and the *conceptual model* is the verbal description and/or mathematical principles that will be used for movement trace generation. The conceptual model is developed through an *analysis and modeling* phase, the computerized model is developed through a *computer programming and implementation* phase, and inferences about the problem entity are obtained by conducting computer experiments on the computerized model in the experimentation phase [Sar98]. The purpose of the mobility model in a network simulation is to serve as a tool, i.e. it is not intended as a way to obtain deeper knowledge on the movement patterns, making the experimentation phase less applicable (contrary to traffic microsimulations which are often used to obtain insights on the traffic conditions under study).

The model is validated and verified by following the different processes (see Fig. 2) [Sar98]:

CONCEPTUAL MODEL VALIDITY Establishing that the theories and assumptions underlying the *conceptual model* are correct and that the model representation of the problem entity is "reasonable" for the intended purpose of the model.

COMPUTERIZED MODEL VERIFICATION Ensuring that the computer programming and implementation of the conceptual model is correct.

OPERATIONAL VALIDITY Determining that the model's output behavior has sufficient accuracy for the model's intended purpose over the domain of the model's intended applicability.

DATA VALIDITY Ensuring that the data necessary for model building, model evaluation and testing, and conducting the model experiments to solve the problem are adequate and correct.

We are not going to discuss the possible approaches to computer program verification as it is related to good programming practices and we are primarily concerned with the conceptual model definition and validation. Furthermore, there exist multiple sources readily available on the subject (e.g. unit testing [otICS99], static verification [ABD$^+$04]).

2.2 DATA VALIDITY

Data validity is an important prerequisite for obtaining a valid model. It is of great importance to physically based synthetic models (presented in Section 3.1.2), and even more so for empirical and data-driven models (see Section 3.2). Unfortunately, the practitioners of mobility models for wireless networks rarely have the opportunity to design data collection experiments, and in most of the time adapt to already existing data sets (as the ones introduced in Sections 3.2 and 3.3).

Figure 1 presents a cognitive map of the types of traces that have been used as basis for the construction of mobility models. Later on,

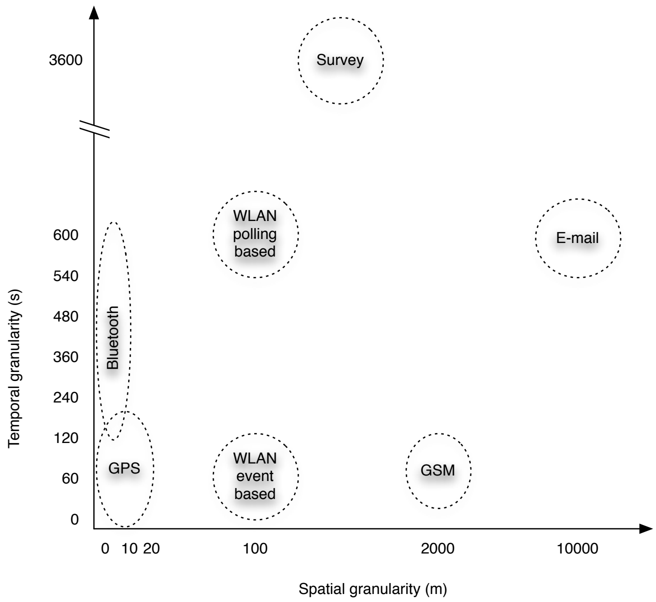


Figure 1: Spatial and temporal granularity of different types of traces.

we discuss in more detail the WLAN (Section 3.3.1), Bluetooth (Section 3.3.2) and GSM (Section 3.3.3) traces. It should be noted that the different types of traces have different degree of coverage, e.g. while GSM networks cover a significant part of a country, typical WLANs rarely cover more than a hundred buildings. In addition, GSM and GPS devices are typically switched on most of the time, while laptops (the predominant WLAN trace generators) are turned off for a non-negligable part of the day. For that reason, WLAN traces are called coarse-grained, whereas GPS traces - fine-grained. Bluetooth traces are particular, as they register only device encounters, e.g. there is no trivial way to obtain the absolute position of the device.

We are not going to focus on data collection and processing issues outside of what we have provided in the the surveys of already used data sets. Instead, we will just point out that standard data treatment and collection validations should be followed, e.g. empirical data should be treated carefully with complete understanding of the limitations of the used data collecting mechanisms and possible side effects (e.g. ping-pong effect in WLAN traces).

2.3 CONCEPTUAL MODEL VALIDITY

This type of validity is needed to assure that the model is built on the correct theories and assumptions, that they are applied as intended, and that the model itself adequately represents the desired entities' movements.

An example of wrong assumptions is the uniform node distribution under RWP movement - it took several years until that assumption was proved wrong (from the time it was introduced in [JM96] to more recent works such as [BRS03, YLN03a, LBV05a]), and in the mean time many papers performed simulations in the non-stationary regime of the model (such as [SJ04, AYS+09]).

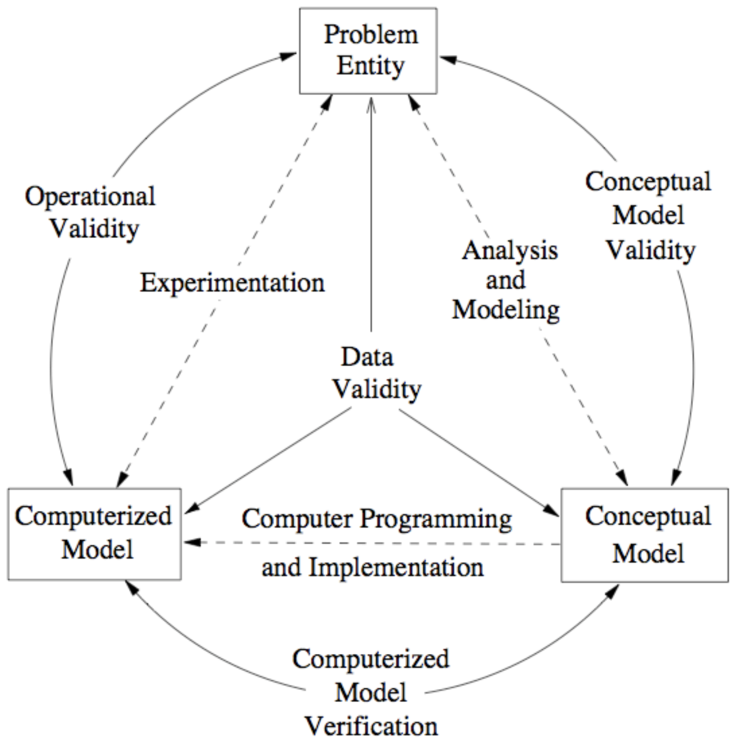


Figure 2: Simplified schema of modeling process. (From [Sar98]).

Unlike the *operational validity*, it is difficult to provide precise metrics with which the conceptual validity to be accepted or rejected. We can point out, however, some guidances that could be used to determine the typical characteristics of the movements of people in general. To this end, we will use the results given in Section 3.3, as well as some additional studies.

People, in general, have a time-variant, highly periodical type of behavior. Generally, they have few frequently visited spots and can be divided into behavioral categories. Behavioral categories are heterogenous (there can exist significant discrepancies between two such categories) and can be represented in a compact way with small loss of accuracy. When considering the microscopic movement behavior, speed is negatively correlated with the density (e.g. when there are too much pedestrians/vehicles a person/car will likely move slower). The speed of a person is also negatively correlated with the size of the group she is in [KSMK05].

2.4 OPERATIONAL VALIDITY

The focus in this paragraph is going to be on *operational validity* (also known as *external validity* [Car96]), which is concerned with determining that the model's output behavior has the accuracy required for the model's intended purpose over the domain of its intended applicability [Sar98]. There are many validation techniques applicable to this type of validity, as summarized in [Sar98]. The methods employed most frequently by mobility model researchers (such as *comparison to other models*, *historical data validation*, etc.) involve comparison of the values obtained with the help of different *mobility metrics* (a survey of the most frequently used metrics is given later in Section 2.5). Indeed, if we know the principal metrics and their target values which completely characterize a given real-world situation, we can easily estimate which scenarios satisfy them, and are thus operationally valid (for the given situations). We are going to call a set of metrics combined with a set of target values (or ranges of values) a *context*.

Furthermore, because model output may change dramatically depending on parameter values, and because different mobility models are sometimes combined we are going to address the ways concrete *simulation scenarios* are validated, where *simulation scenario* is defined as a set of mobility models, their corresponding parameters and all relevant simulation settings (duration, area dimensions, border behavior, etc.). Note that a given scenario may include multiple simulation setups, which differ only in the mobility models' parameters.

Hereafter, a *simulation scenario* (models + parameters) will be considered *valid* only in a given *context* (metrics + target values).

In order to improve the acceptance of a given context, it should be accompanied with a short description and a list of tags characterizing the real-world situation it represents (e.g. "people movement, beach, high-season, weekends"). It should be noted that ideally the definition of a context should be done by an expert in the field, before (or independently of) the development of a mobility model or simulation scenario.

Finally, as an example we are going to provide the definition of the *context* for universities or corporation environments. Some (or even most) of these points can be extended to more general contexts (such as city workers). However, as there have not been no conclusive research on the subject, we are going to limit our presentation to the contexts that have been thoroughly studied. From the various studies detailed in Section 3.3, we can summarize the following metrics and accepted values for university and corporate contexts:

- Power law or BiPareto distribution for the aggregated Inter-Contact Time (ICT) over all node pairs.
- Log-normal distribution for pairwise ICT. Eventually exponential or power-law are also accepted. In addition, the various distributions should not follow the same parameterization (there should exist a heterogeneity).
- Contact duration - power law.
- Power law or BiPareto distributed session length.
- Power law distribution of dwell time.
- Well defined home location at which every node should pass a significant fraction of its time.

The example provided here does not specify more details because the selected context is very broad. More metrics and parameters can be defined, and the type of distribution can be determined with greater accuracy, if the context is limited to a concrete case, e.g. the shape parameters of the distributions can be given for many universities whose networks were studied (see Section 3.3). Additional restrictions can be formulated, e.g. by restricting the movement dynamics (speed, acceleration, etc.) in order to further specialize the context (e.g. pedestrians in university).

2.5 MOBILITY METRICS

Here, we are providing an overview of the most popular metrics used for measuring the characteristics of mobility models. An in-depth survey and formal mathematical representation framework for MANET metrics may be found in [XBJ07].

Measurements could be taken at a single node (*node-centric*) or could be calculated for the whole set of nodes (*network-wide*) - both methods have their advantages and applications. Node-centric measurements could be easily performed at runtime on a device taking part of a deployed solution. The results could be used to fine-tune the behaviour of the node, e.g. as in [JBC02] or [AUS01] where the authors change the behavior of the routing protocol based on the *link duration* or *link change rate* respectively. On the contrary, network-wide metrics are much more difficult to calculate in real-world networks. It is not impossible, as each node may record appropriate node-centric data which could be then analyzed in a centralized manner, but resource constraints and increased complexity make this type of usage less likely. However, network-wide calculated metrics can give invaluable insights of a mobility model as a whole and can thus indicate the corresponding real-world scenarios (or vice-versa). This makes the network-wide measurements extremely useful during the design phase of a solution.

The metrics found in the literature could be classified according to the granularity of the studied trajectories into two groups - *coarse-grained* (e.g. WLAN traces) and *fine-grained* (computer generated traces, geolocation traced, etc.). Almost all metrics can be measured in both node-centric and network-wide context, so the descriptions hereafter will be given in the former case, whenever possible. We will not give the way average values are obtained from the individual measurements because in most of the cases this is a simple average over all nodes and for the entire simulation. An alternative approach to analyzing network-wide behavior is to investigate the distribution of the values of the metrics for the whole simulation (as seen in [SBKH03]) an approach frequently used for encounter metrics, which can also be applied to all other metrics.

2.5.1 *Fine-grained trace metrics*

Because the almost exact location of each node can be found for any given moment, there are much more aspects of the movement patterns that could be explored. Zheng et al. described two categories [ZHR04] - *direct* and *derived* mobility metrics. The former evaluate phenomena with clear physical correspondence (such as speed or acceleration), while the latter use mathematical modeling to measure the changes to some logical structures (e.g. connectivity graph).

2.5.1.1 *Direct metrics*

a. **Node pause time** [DPR00] - The average time a node remains stationary for the whole simulaton.

b. **Node movement time** - The average time a node is not stationary.

c. **Move-stop ratio** [HMS+05] - Total movement time divided by the total stationary time.

d. **Node density** [JBRAS03] - number of neighbors per node. The distance in which a node is considered as a neighbor is given as parameter. Most commonly this parameter is chosen to correspond to the transmission range R.

e. **Speed** - the momental speed of a node.

f. **Mobility factor** [LH98] - Average change in distance between a given node and all other nodes for a given moment.

g. **Relative speed** [JLH+99a, BSH03] - For a pair of nodes, the speed of the first, relative to the second.

h. **Degree of spatial dependance** [BSH03] - the extent of similarity of the velocities of two nodes that are not too far apart.

i. **Displacement measure** [HKG+01] - The difference between the actual distance traveled and the geographic displacement of a node for a period of time. If a node starts moving and for the given time period returns to its departure point, the actual distance traveled would be the entire path it moved, while the geographic displacement would be zero.

2.5.1.2 *Derived mobility metrics*

a. Connectivity graph metrics

   These metrics measure the evolution of the connectivity graph.

   a) **Link change rate** [HGPC99] - the sum of all new and all severed links for a node at a given moment.
   b) **Number of Link Changes** [BSH03] - The number of times the link between two nodes transitions from "down" to "up".
   c) **Link duration** [JBC02, BSH03, SBKH03] - The average duration of the link existing between two nodes.
   d) **Normalized unreachable pairs** [SJ04] - The ratio of the total number of *unreachable pairs* to the total number of possible pairs at a given time, where two nodes form an unreachable pair if there exist no path between the two.
   e) **Path length** [SJ04] - The minimal number of wireless hops between two nodes at a given time.
   f) **Route change rate**
      - **Minimal shortest route-change** [HJ00] - The minimum number of times a pair of nodes would need to change routes in order to always have a shortest (least wireless hops) path.
      - **Minimal route-change** [HJ00] - The minimal number of times a pair of nodes would need to change the route between them, where a route is changed only when it breaks.

g) **Path duration** [SBKH03] - The time a specific path between two nodes exists. The shortest path between two nodes is selected, and its duration is determined. Even if a shorter path appears (due to network evolution) it is not counted until the original "shortest" path is broken.

h) **Path availability** [BSH03] - The fraction of time during which a path is available between two nodes i and j.

i) **Clusterhead change rate** [HGPC99] - The number of clusterhead changes in at a given instant. This metric requires an algorithm that groups all nodes into *clusters*, and for each cluster elects a node to be its *clusterhead*.

b. Contact-based metrics

Used extensively for delay-tolerant networks. An *encounter* is the time of incidence and the duration when two nodes are in communication range. In [KMR05] the authors defined a *contact* to be the set of all encounters between two nodes. However, later usage of the term made it equivalent to *encounter* [CHC+05], which is the denomination we will be following.

a) **Contact rate** [KMR05] - The number of new contacts experienced by a node per unit of time.

b) **Encounter frequency** [KMR05] - The number of encounters experienced by a node per unit of time, divided by the contact rate for the same period.

c) **Encounter rate** [KMR05] - The number of new encounters experienced by node per unit of time.

d) **Contact duration** [CHC+05] - The duration of a single contact.

e) **Inter-Contact Time (ICT)** [CHC+05] - the time gap between two consecutive contacts of a node pair.

- Average Inter-Contact Time (ICT) - The average time gap of all ICT over all pairs of nodes.
- Pairwise average Inter-Contact Time (ICT) - The average time gap of ICT of a given pair of nodes.

c. Others

a) **Entropy-based mobility metric** (network-wide) [TBH05] - A mobility metric using entropy to measure the topological uncertainty of a network.

2.5.2 *Coarse-grained trace metrics*

1. **Activeness of users** [HH05] - the tendency of a user to be online measured by how actively the user shows up in the trace.

2. **Macro-level mobility of users** [HH05] - how widely a node moves in the network in the long run, and how its online time is distributed among the Access Points (APs).

3. **User daily diameter** [KKK06] - The maximal distance between the visited APs by a given user during a single day.

4. **Session diameter** [HKA08] - The maximal traveled distance without interrupting the association to the network (e.g. roaming between APs is allowed).

5. **Dwell time** [TP06] - the time a user spends in a given location without leaving it (e.g. associated to an Access Point (AP)).

2.6 CONCLUSION

In this chapter we argued that even if the informal notion of model realism can be defined in intuitive manner, it can be difficult to perform a rigorous study without a formal understanding of its nature. This is supported by studies finding purely synthetic, memoryless models as the RWP to model real world situations [SJ04, RBA05].

Instead of subjectively defining a model as being "realistic" we preferred discussing only whether a model is valid and verified. Model verification refers to the correctness of the way the model was implemented, e.g. "are there any bugs in its computer realization?". Model validation can refer to several types of validity, which we outlined in this chapter.

The conceptual model validity is fundamental, and the first one to be confirmed. The reason for this is because it assures the correct assumptions and theories are used and without it all other types of validation can be biased.

Data validity refers to the correctness and the adequacy of the data used for the definition of the model. We have provided a cognitive map classifying the various kinds of data used as basis for mobility models seen in the literature.

Finally, we based the definition of mobility model validity upon the concept of operational validity, which consists of determining if model's output behavior is "reasonable". More formally, a mobility model is defined to be *valid* in a given *context*, where a context is a set of metrics and target values. Even though model validation is a commonly performed operation, this is the first time model validity is defined in a formal way for mobility models. In addition, we have briefly surveyed the most cited mobility metrics which can be used for model context definition and validation.

The concept of "realistic" model can now be replaced with the formal assertion that a model is valid in a given context. Our purpose will now consist of defining such models, which also possess the major mobility model characteristics outlined in the introduction (Chapter 1). However, before proceeding with this goal, we will present a survey of the existing work in the next chapter (Chapter 3).

3 EXISTING APPROACHES

In this chapter are presented some of the most important models and trace analyses related to microscopic movement generation and analysis for wireless network studies. The major research directions pursued by the wireless networking community are determined mostly by the type of network being simulated, including Wireless LAN (WLAN), Delay/Disruption Tolerant Network (DTN), Mobile Ad-Hoc Network (MANET), Vehicular Ad-Hoc Network (VANET), Wireless Sensor Network (WSN), Cellular networks, Satellite networks and Underwater networks.

Each of these types of networks has a typical field of application and deployment, and determines the type of movement entities that has to be modeled. Typical entities which are used for simulation of these networks include, but are not limited to:

1. Animals - DTN, WSN
2. In-body movements (e.g. nanorobots) - MANET, DTN
3. Pedestrians - WLAN, MANET, Cellular networks
4. Bicycles - WLAN, MANET, Cellular networks
5. Vehicles - WLAN, VANET, Cellular networks
6. Trains - WLAN, Cellular and Satellite networks
7. Airplanes - Cellular and Satellite networks
8. Marines and Submarines - Underwater and Satellite networks
9. Satellites and spaceships - DTN, Satellite networks

Every one of these mobile entities has its specificities and requires dedicated investigation in order to define corresponding *contexts* of validity and fundamental movement principles. Most of them are very specialized (in-body movement entities, marines and submarines), or exhibit strongly constrained and centrally regulated movement (trains, airplanes, satellites, spaceships) and as such are usually simulated with specially developed models. The movements of the rest of the entities, however, are not regulated and exhibit many degrees of freedom. Incidentally, WLAN, MANET, VANET, WSN, DTN and Cellular networks are comprised mainly of these types of entities, which makes their movement patterns of a particular interest.

There exist two main approaches to modeling the movement patterns of these these entities. The first, aimed at defining easy to analyze synthetic models, is founded on the grounds that it is very difficult, if not impossible, to obtain complete real-world traces for most of the entities, and as such only general movement characteristics should be simulated. The second, based on heavy collections of empirical data, tries to extrapolate the movements from data traces, even tough the data may have been collected with entirely different purpose.

These approaches can be further subdivided into the following categories:

1. Synthetic Models.
   - Analytical - based on purely mathematical or æsthetical[1] considerations.
   - Physically-based - models following established principles or characteristics of the simulated movement (e.g. bio-inspired).
2. Empirical/Data-driven Models - the characteristics of the model are determined by real-world data.
   - Coarse-grained trace based (cellular network traces, WLAN traces, surveys, ...).
   - Fine-grained trace based (Global Positioning System (GPS) traces, video recordings, etc.).
   - Encounter trace based (such as Bluetooth contact traces).
   - Map based.

A model may be validated for certain situations independently of the principles upon which it was built (e.g. a synthetic model may fit the empirical observations). However, the type of a model is evident mainly from the way it was created, e.g. a synthetic model which is only later shown to match some empirical observations is generally going to be different than an empirical model proved to possess some mathematical properties.

The models that are going to be presented in this chapter are summarized in the cognitive map shown in Figure 3. It illustrates the differentiation between the models by ordering them in a logical order by the complexity of their configuration (e.g. number of parameters). The map is compiled based on typical usage scenarios and is by no means a rigid organization (e.g. for a 1s simulation of a single node, configuring a map-based model with the map of an entire country is more complex than specifying the complete traces of the node in question). Instead, it places the models in a relative relation frame which should be used as a reference point on which can be based further comparisons. The growing number of configuration data and parameters is in direct relation to the specialization of a model - starting from the most general models which do not have any specific real-world correspondence (i.e. context-less) to the complete trace movement description where each node position is given for every second of the simulation (i.e. context-full). The scheme emphasizes the tradeoffs that are made when choosing a model in respect to its complexity, configuration difficulties and real-world context specialization.

3.1 SYNTHETIC MOBILITY MODELS

3.1.1 *Analytical Models*

Some of the following models have been present in the predominant part of the surveys on mobility models appropriate for MANET or VANET simulations. Ever since Davies [Dav00] and Camp et al. [CBD02] categorized these models into *individual* and *group* movement patterns,

1 Æsthetic - concerned with beauty or the appreciation of beauty [McK05]. In the context is used in the sense that synthetic analytical mobility models are typically simple, following few equations with few parameters, e.g. possessing some kind of minimalist and/or mathematical beauty.

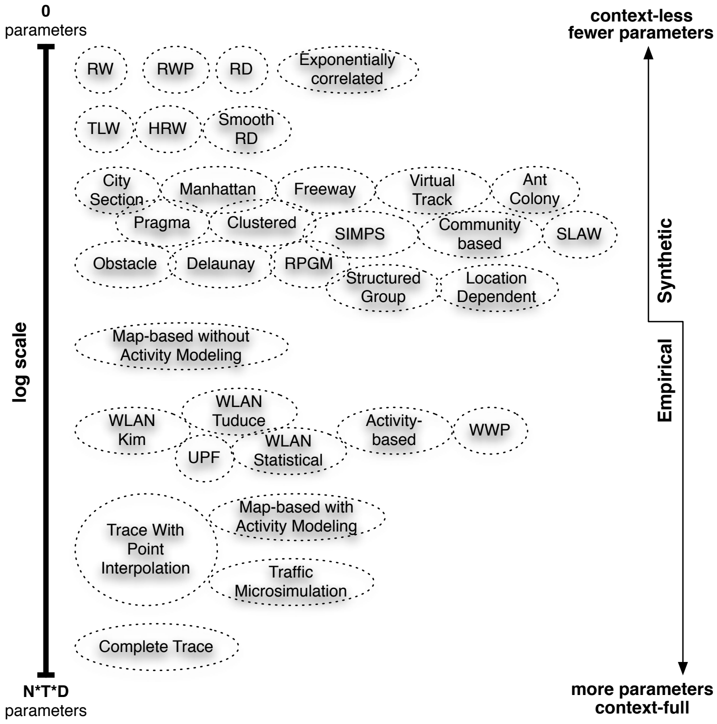


Figure 3: Outline of several mobility models ordered according to their parameter complexity (note the logarithmic scale). The maximal complexity is reached with a complete trace description for a T-second simulation of N nodes in D dimensions. The models were ordered according to their typical usage scenarios (e.g. $N, T \gg 1$).

they have become the most cited and presented models in the mobility-related literature. Other papers [SI01,BH04,ZHR04,Sch06] also surveyed these simple, purely synthetic models, with some minor additions.

3.1.1.1 *Random Waypoint (RWP)*

The movement of each node is divided into epochs of movement and pause. During the pause epoch, the node does not move. In the beginning of a movement epoch, the node chooses a destination point and a speed, and starts moving towards the selected point with that speed. In the classical RWP model the pause time and the speed are drawn with uniform distribution from fixed intervals. The path generated by a single node following the RWP is shown in Fig. 4.

This model is by far the most used one for MANET simulations (a claim supported by [KCC05]). Indeed, ever since it was introduced for wireless network simulations in [JM96], it has become the basic model which comes integrated in all simulators used for MANET research, its description is short and concise, it has very few parameters, and they are all easily understandable. Furthermore, it is one of the most studied models, with most of its relevant properties well known and understood, such as the connectivity of the network formed by nodes moving according to it [LHK04], the topology change rage [PCBH03], and so forth.

Its simplistic nature, however, hides some surprising particularities that when neglected could lead to false results and that even made some of the researchers to declare this model as harmful [YLN03a]. These problems are due to two seemingly natural assumptions, which do not hold in the majority of cases. In [LBV05a] the authors show that the "intuitive" interpretation is wrong because of the difference between time- and event-averages.

An important remark on the RWP is that although it has a steady-state regime in which node distribution is a stationary random element, its stationary distribution depends on the simulation area. The first assumption that is not always true is that the steady-state node distribution is uniform. In fact, it is such only in some rarely-used cases, e.g. when the simulation area is a sphere [LBV05a]. However, for the most frequently used case, when the simulation area is a 2D rectangle, this is not the case. The reason for this "anomaly" is that when a node is closer to one of the borders, the probability of selecting a destination waypoint with direction "towards the center" is higher. Moreover, when moving to a point situated on the other side of the center, the trip passes through the center. Those two factors result in an increased node density around the center and decreased one around the borders.

The second assumption is that the average speed of all nodes in a given moment would match the average of their speed distribution. However, because a trip ends sooner if the node moves faster, the nodes gradually become trapped in long trips at low speeds, thus the average speed decreases until reaching the stationary level (Fig. 5). Further, if the minimal movement speed is zero, the model will approach asymptotically a static scenario (illustrated in Fig. 6), never reaching a steady-state [YLN03a, LBV05a].

Several studies have shown that, although the density is not uniform and the average speed is not the average of the speed density, the model stabilizes (reaches its steady-state) after an initial warm-up period. A simple solution to the aforementioned issues is to discard the first

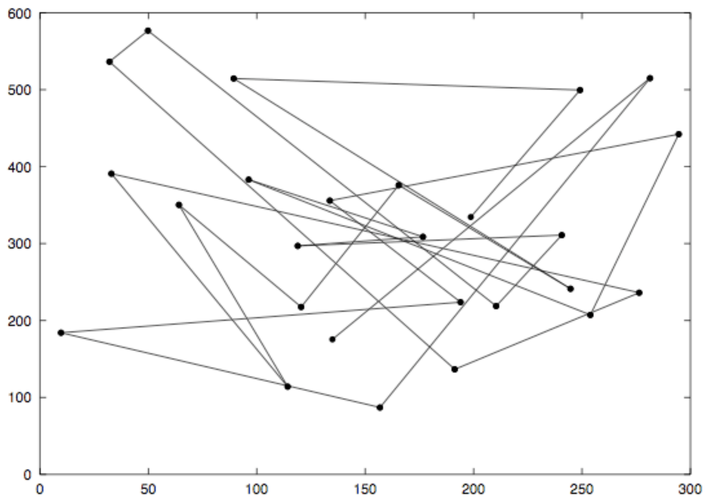


Figure 4: Trajectory of a node moving with the RWP movement pattern in a 2D area (From [CBD02]).

900 iterations of the simulation and to set a positive minimal speed [YLN03a] ([CBD02] recommends discarding the first 1000 iterations). A better solution is to use the model's steady-state speed and spatial distributions during the initialization phase [Nav04, LBV05a] as in some situations it may take longer than 1000 iterations for the model to reach its steady state. However, the closed form of the steady state distributions depend on several factors (the type of the simulation area, and the pause time and speed distributions) and in the general case include involved calculations (as in [BRS03, Nav04, LBV05a, HLV06]). In [LBV05a] the authors provide an algorithm sampling the initial conditions of the simulation which does not require the calculation of the exact distribution. Other possibilities include running several long simulations, saving the simulation states after the simulation has stabilized, and then using the saved states for initialization [CBD02], or using initialization bias removal procedures [LC02].

Another objection to the usage of the RWP is that it does not represent any realistic movement and may lead to wrong conclusions. Even though there exist real-world scenarios that can be faithfully represented by it [RBA05, SJ04], it has been shown that the traces produced by the RWP do not possess the characteristics reported by multiple studies, notably the truncated power-law distribution of the inter-contact times, and the small-world property of the graph of the interactions.

3.1.1.1.1 SCENARIOS AND MODIFICATIONS

The RWP has been used as a basis for the creation of multiple network simulation scenarios. Most of the theoretical frameworks include sections dedicated to its realization in the context of the framework at hand, e.g. [YLN03b, LBV05a, AS07a, AS07b]. It has been extended in various ways, notably using distributions other than the uniform for the speed and pause selection [BRS03], including making some of these distributions dependent on the current location, or a Markov chain.

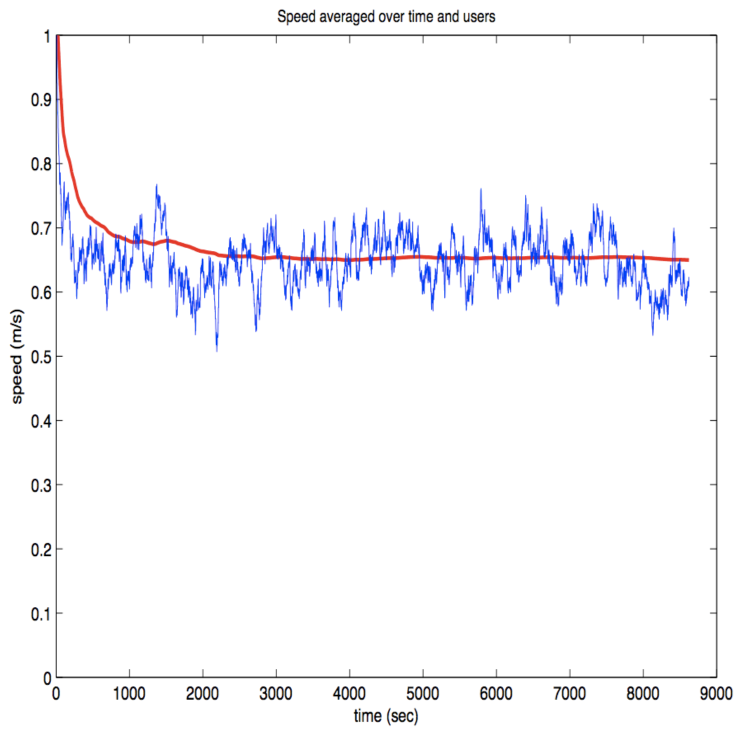


Figure 5: Speed averaged over time and users (thick line) and instant speed averaged over users (thin line). There are 200 independent nodes, with the speed selected uniformly between $v_{min} = 0.1m/s$ and $v_{max} = 2m/s$. The simulation area is a square, with side $1000m$ (From [LBV05a]).

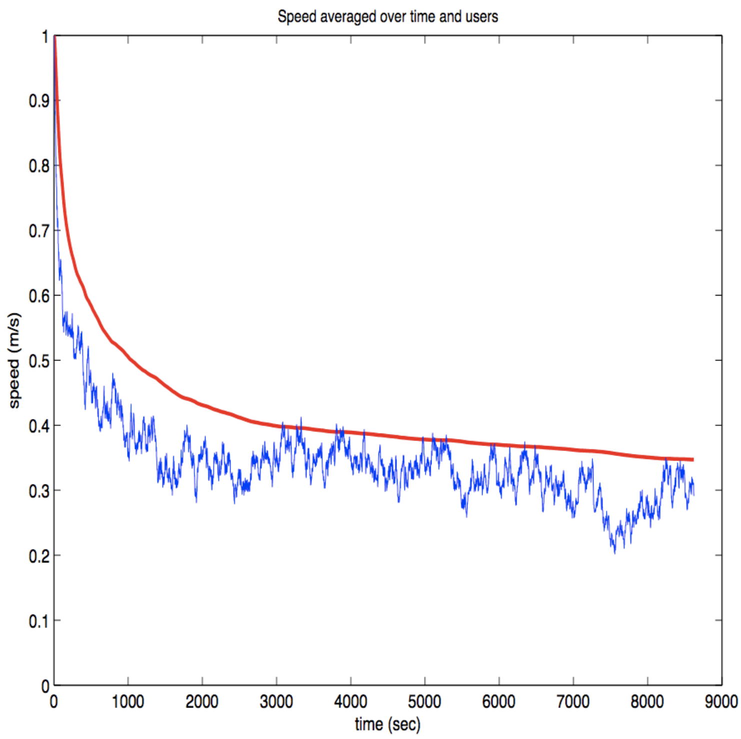


Figure 6: Same as Fig. 5, but with $v_{min} = 0m/s$ (From [LBV05a]).

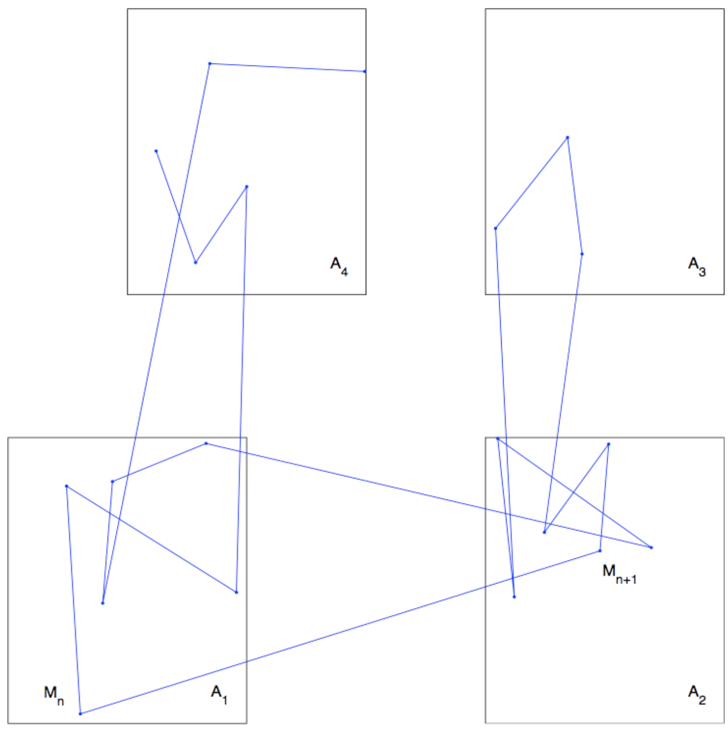


Figure 7: Model of the simulation, with four towns $A_1 - A_4$. M_n is the n^{th} position of the node (From [LBV05a]).

In [BGB01], the authors propose the *Restricted* RWP, which simulates the movement of nodes between several towns. With this model, the node moves according to the RWP in a rectangular region (representing a city) for a selected number of moves, after which it moves to another city connected to the current one with a highway (Fig. 7). In [LBV05a] the authors provided the stationary distribution of this model.

In [AFMT04] the authors used the RWP to create a rescue mission scenario. The scenario simulates the method called "separation of the room", where the area is divided into different areas: *incident site*, *casualties treatment area*, *transport zone*, and *hospital zone*. Each of the areas is modeled with a square (some of which overlap), with the nodes moving according to the RWP within the square.

3.1.1.2 *Random Walk (RW)*

The random walk, introduced by [Gue87] for wireless network simulations, is the discrete version of the *Brownian Motion*, first quantified by Einstein [Ein05]. A node moving according to the RW chooses a direction in which to move (uniformly in the unit circle) and then moves in that direction with a randomly selected speed for either a fixed interval of time, or a fixed distance. The speed is chosen with uniform probability in $[v_{min}, v_{max}]$. Illustration of this movement is given in Fig. 8.

The RW is a memoryless mobility model (i.e.. its current state does not depend on the past), which leads to sharp changes in the movement direction. It has be proven that on a two-dimensional lattice (e.g. a rectangle), a node moving according to the RW will visit any point

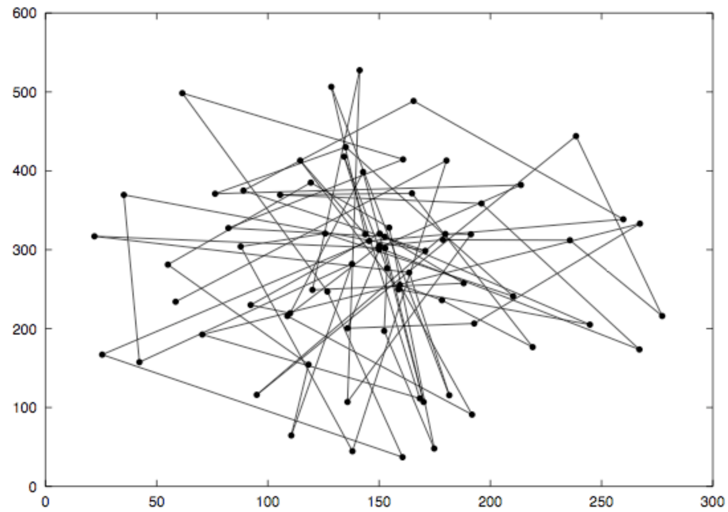


Figure 8: Trajectory of a node moving with the RW movement pattern in a 2D area (From [CBD02]).

(including its starting point) if given enough time (or, as the movement time approaches infinity).

Similarly to the RWP, the RW is a simplistic model, available for all simulators, and with extensively studied properties (such as [HMT08, BRS02]). One of the major differences between the two is that because the Brownian Motion is a fundamental stochastic process used in all fields dealing with non-deterministic system behavior (economy, biology, meteorology, . . .), it is more thoroughly understood. Furthermore, because with reflection or wrap-around border behaviors the stationary distribution of the node density is uniform [LBV05b, BRS02], the most used RW scenarios do not suffer from initialization bias. Because of its mathematical tractability, the RW is extensively used for cellular network studies, both analytical and using simulations, e.g. [ZD97, VF99].

Although in this form the RW produces exponential ICT distribution considered to be non-realistic if characteristic for all pairs, [KLBV07] showed that in some cases a RW may produce power-law ICT distribution with exponential tail (as seen in real-world traces). Furthermore, in [CE07] the authors showed that the exponential ICT are an artifact of the simulation boundary, and that by choosing infinite, or sufficiently large simulation area the classical RW also produces power law (with exponential tail) distribution of ICT.

3.1.1.3 *Random Trip*

The Random Trip, defined by Le Boudec and Vojnović [LBV05a, LBV06b], provides a theoretical basis for creating stationary mobility models. It is founded on the Palm calculus, which relates time averages to event averages. It can be considered as a framework, or mathematical theory applied to the mobility modeling, and is defined as a set of specific conditions that have to be fulfilled by a mobility model in order to be declared Random Trip. Provided that this is the case, the model is guaranteed to possess a stationary regime.

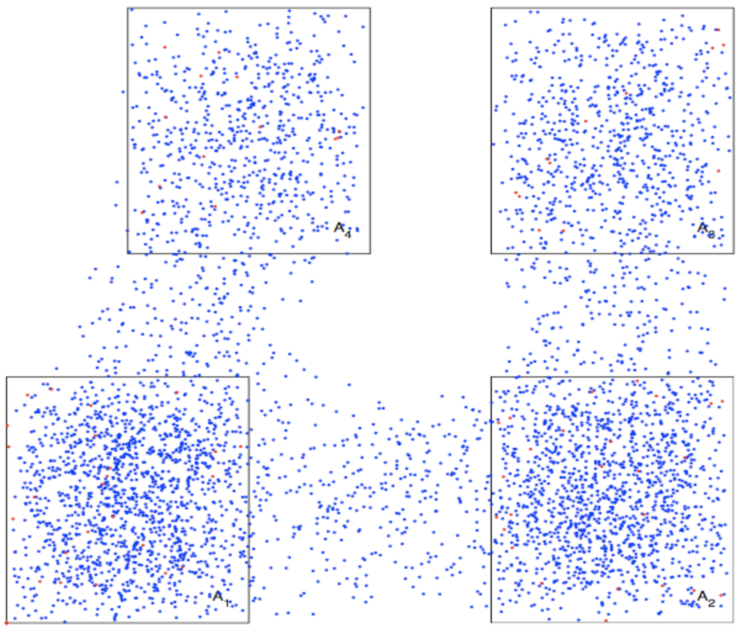


Figure 9: Node positions sampled from the stationary distribution of the Restricted RWP. (From [LBV05a]).

Models which fall in this category include RWP, RWP on general connected domain, Restricted RWP, RWP on a sphere, and RW with wrapping or reflection [LBV05a].

Even though some of the conditions are straightforward, others are more elaborated and may require considerable investigation. Indeed, the requirements of the Random Trip are equivalent to proving the existence of a stationary regime, so other approaches may turn to be more appropriate for a particular model.

3.1.1.4 *Graph-constrained*

Constrained movement is one of the characteristics frequently put forwards as a requirement for the realism of a mobility model (as in [HFB06]). The wide majority of mobility models use a graph embedding in the simulation area, with the movement of the nodes restricted on its edges and vertices. The graph may be directed in which case nodes may only move in the direction of the edges.

This type of constraint may be used to successfully model the road network of a city, the streets linking several houses, the highways connecting two cities, and any other environment where node trajectories are naturally restricted to frequently used pathways. Indeed, this is the case for most of the places equipped with an existing infrastructure. Here, we are will give the several examples of such models.

3.1.1.4.1 GRAPH-BASED

The model from [THB$^+$02] restricts destination point selection to graph vertices, which are drawn with uniform probability. It uses shortest path to move towards the selected destination and after reaching the

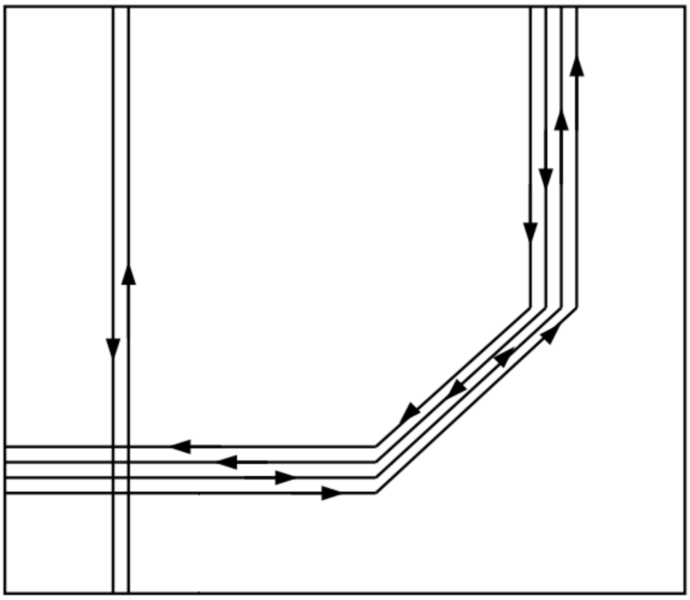


Figure 10: Trajectory of a node moving with the Freeway movement pattern in a 2D area (From [BSH03]).

destination vertex, it pauses. It may be regarded as performing a RWP movement restricted on a graph.

3.1.1.4.2 FREEWAY

The *Freeway* Mobility Model introduced in [BSH03], follows a directed graph in the shape given in Fig. 10. The speed of a node depends on its previous speed and cannot exceed the speed of a preceding vehicle if it is less than a safety distance away. Once a node is placed on a lane it cannot change it.

3.1.1.4.3 OBSTACLE

In [JBRAS03] the authors generate the constraining graph by first placing buildings on the simulation area, and then calculating a Voronoi diagram on their corners, which produces the constraining planar graph (Fig. 11). The nodes move according to RWP constrained on the graph, that is, a node selects a point on the graph, moves following the shortest path, and after reaching the destination, pauses.

Because of the way the Voronoi diagram is specified, some of the edges traverse the buildings, and when they do, it is in the middle of the building wall.

The model was further extended in [JBRAS05] with the explicit placement of doorways and exponentially distributed destination point selection.

3.1.1.5 *Group*

There exist several models aimed at modeling group movement patterns. These models are based on the observation that often people move in groups, a trait rarely seen in the other models. Indeed, having several

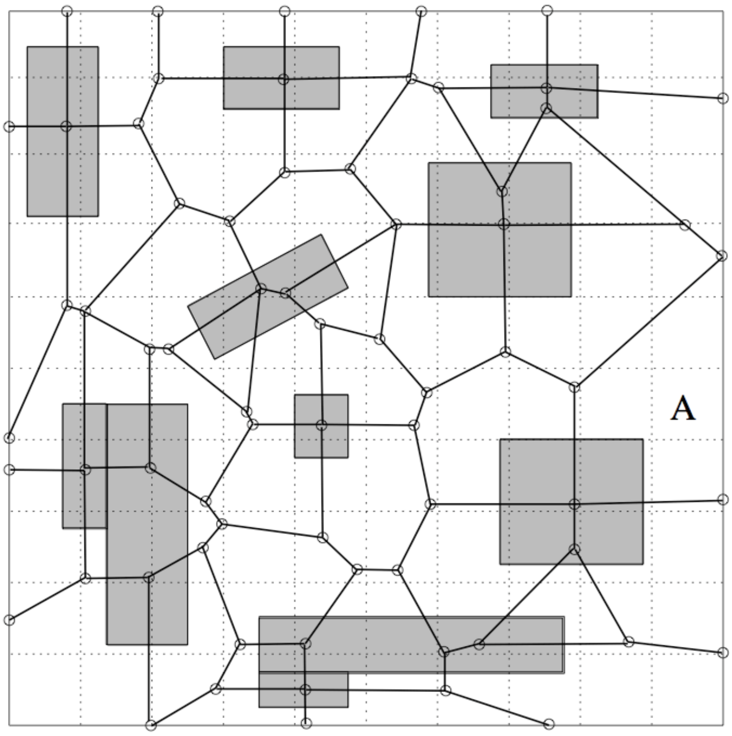


Figure 11: Planar graph of the Obstacle mobility model (From [JBRAS03]).

nodes to move together toward a common goal requires in most cases explicit specification. Moreover, group movements have important implications on the neighbor stability and topology change rate, which may dramatically affect routing protocol performances. It should be noted, that along with a group model, an appropriate traffic model should be used.

3.1.1.5.1 REFERENCE POINT GROUP

The Reference Point Group Mobility (RPGM) introduced in [HGPC99] is one of the most cited and general group mobility models. Each group has a logical center, which defines the movement of the entire group, and each member of the group is given a reference point which is always fixed with respect to the group center. When the group center performs a movement with vector v, the reference points of all nodes of that group also move with the same vector. The nodes, however, have some degree of randomness around their reference point - at each step, every node moves to a random position in the vicinity of its reference point. The vicinity is defined as a circle with radius r.

In [HGPC99] the group centers follow a manually specified trajectory (Fig. 12), but it is possible to imagine a scenario where the movement of the group center is governed by another mobility model (such as the RWP).

It is possible to obtain other group models by choosing appropriate reference point layout. Example models include the *Nomadic*, *Pursue* and *Column* mobility models [San]. The *Column* model the scenario of walking column of solders, the *Pursue* models a group tracking a particular node, and the *Nomadic* - group movements where every

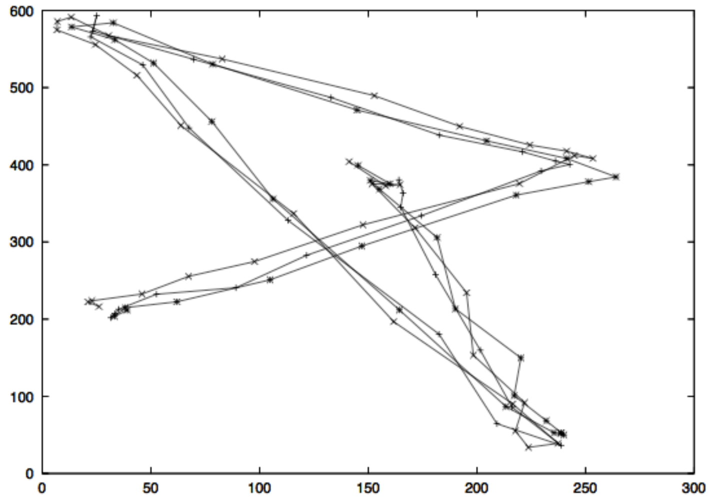


Figure 12: Trajectory of a single group with three nodes moving with the RPGM movement pattern in a 2D area (From [CBD02]).

member has its own personal moving space (that is, the vicinities around the individual reference points do not intersect).

The *Reference Velocity Group Mobility Model* [WL02] is another model which can be modeled with RPGM. It uses group velocity instead of logical center, and can be considered as a different approach to obtaining the same movement patterns.

An important drawback of this model is the static group structure. A node cannot leave nor join a group. Groups cannot be formed or dissolved during the course of the simulation. This rigid behavior may be suitable for certain scenarios, but is not apt for more general simulation cases.

3.1.2 *Physically-based Models*

3.1.2.1 *Heterogeneous*

The Heterogenous Random Walk (HRW) defined in [PSDG08, PSDG09b] is based on the RW, but was modified to include correlation between speed and the inverse of the node density at a given location, and the spatial stability of clusters in the connectivity graph over time.These two characteristics were discovered by the authors in several GPS trace sets [cab, spo]. The model adds heterogeneity to the distribution of the speeds, and as a consequence, of node density.

The simulation area is a torus, and is divided into two regions, each of which possessing an individual speed range. More precisely, the variance of the speed in each of the regions is given as parameter (Fig. 13). The nodes execute a standard RW, with the modification that the selected speed depends on node's current region speed range. The authors studied the mathematical properties of the model and proved that it has a stationary regime. Additionally, a closed-form expression of the time-stationary node distribution if provided. Finally, the authors have described an algorithm for performing a perfect simulation, i.e. starting the simulation from the steady state distribution.

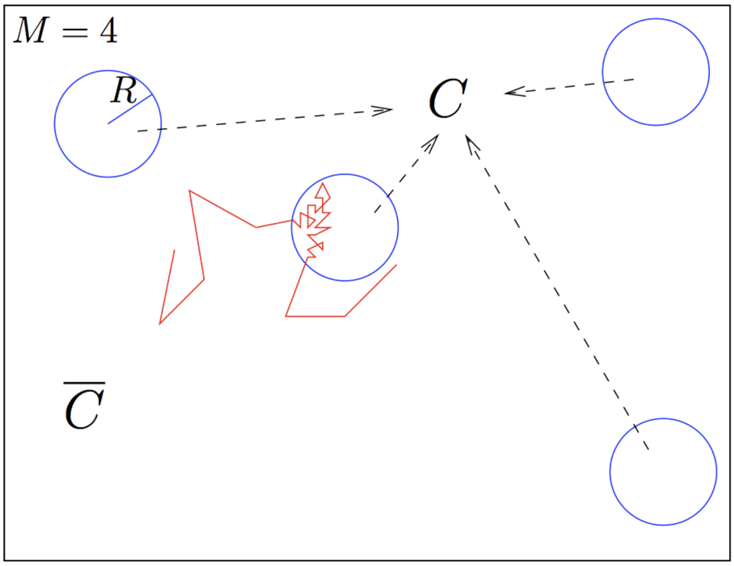


Figure 13: HRW - Nodes move slower inside the area C than inside area $\bar{C}$, with C constituted of $M = 4$ circles with radius R (From [PSDG08]).

HRW is simple, elegant mobility model, with good stochastic behavior (e.g. it has a stationary regime) and is easy to understand. It has been validated against realistic metrics they described in the paper. However, the authors have not provided evidence about the realism of other widely accepted metrics, such as the inter-contact time and contact duration distributions. Because it is based on the RW, it is not clear to what extent the sociological interactions between the nodes are going to be realistic, e.g. it is unlikely to generate emergent group behavior. These properties need to be studied, because even though the model has validated the link between node speed, node density and the existence of spatially stable clusters in the connectivity graph, these characteristics may be easily modeled with various other approaches (e.g. with the Restricted Random Waypoint model) and it is important to know which one provides a wider array of realistic behaviors.

Finally, the speed/density relationship is based solely observations from vehicular traces and as such may not be indicative for pedestrian movements. In fact, if we consider only human walking patterns, the speed will be almost constant for all walkers and would not be sufficient to explain the dynamics of human interactions. In this case, HRW will become equivalent to RW.

3.1.2.2 *SIMPS*

The Sociological Interaction Mobility for Population Simulation (SIMPS) (Fig. 14) presented in [BLDAF09, BLdAF06] is a model based on the behavioral mobility paradigm introduced in [LBdAF06a, LBdAF06b]. In this framework a set of rules, represented as attractive or repulsive forces, determine an acceleration vector which constantly affects the velocity of the node.

SIMPS uses two types of behavior - *socialization* and *isolation*. When a node is in a socialization phase, it will be attracted by the other nodes, while during the isolation phase it will be repulsed by them. The degree of attraction/repulsion depends on the distance between the nodes (e.g. closer ones have a greater impact than those far away) and the degree of friendship between them. The friendship relations are given with a weighted, directed graph, called the social interactions graph, where the vertices are the nodes, and the edges - the friendship relations. The weight on the edges is a number in $[0; 1]$ where an edge $i \xrightarrow{1} j$ indicates that the node i considers j a best friend, while $j \xrightarrow{0} i$ means that i is completely unknown to j.

A node choses whether it wants to socialize or isolate based on the number of its neighbors. In the beginning of the simulation, each node is given an ideal number of neighbors, and a tolerance interval in which the node feels comfortable. When the node is surrounded by a number of nodes which falls in its tolerance interval, it is in socialization phase, otherwise it tries to isolate itself. The number of neighbors also indicates the intensity with which the node would try to socialize/isolate (called node excitement). The excitement is a value in $[0; 1]$, being 0 (the node doesn't want to change anything) when the node is surrounded with its ideal number of neighbors, and 1 (the node wants to change as fast as possible) when its outside its comfort range.

Additionally, SIMPS uses an imperfect perception model to detect the number of neighbors, which takes the true number of nodes closer than a given radius (which is the classic disk model), but reports the average of the true number, and the last known neighbor count (which was also calculated in the same manner). On initialization, the neighbor count is fixed to be the ideal number of neighbors for the node.

After the behavioral rules have calculated the direction $\overrightarrow{d}$ in which the node should move, and its excitement e, the new desired acceleration is generated by using node's proper maximal acceleration a as $a \times e \times \overrightarrow{d}$. Finally, the current velocity is changed with the desired acceleration, and the new velocity is corrected not to exceed the maximal allowed velocity of the node.

The model is based on observations found in several studies in sociology, which relate to person's socialization desires. The first aspect, *intrinsically*, characterizes the fact that sociability levels of people depend mainly on their age and social class (and not their current situation, for example). This aspect is modeled by fixing the ideal number and the tolerance interval of neighbors. The second one, *interactivity*, indicates that people take actions to fulfill their sociability levels. This is reflected by the definition of two behaviors - socialization and isolation - which make the node to take actions whenever it is not in its ideal environment. Finally, the imperfect perception model is also based on real-world observations.

The authors make extensive parameter space exploration, and show that most parameter combinations create traces with realistic contact duration and inter-contact time distributions (i.e. following a power-law distribution with an exponential cut-off). An interesting finding is that the social graph does not seem to have an important role, as different kinds of graphs produced the same type of distributions.

SIMPS is a simple mobility model, which uses principles of the human social dynamics discovered in several sociology studies. It is promising in that it produces realistic contact and inter-contact distributions. However, no other metrics have been studied, and it is unclear if it generates other realistic characteristics (such as node pause times). Fur-

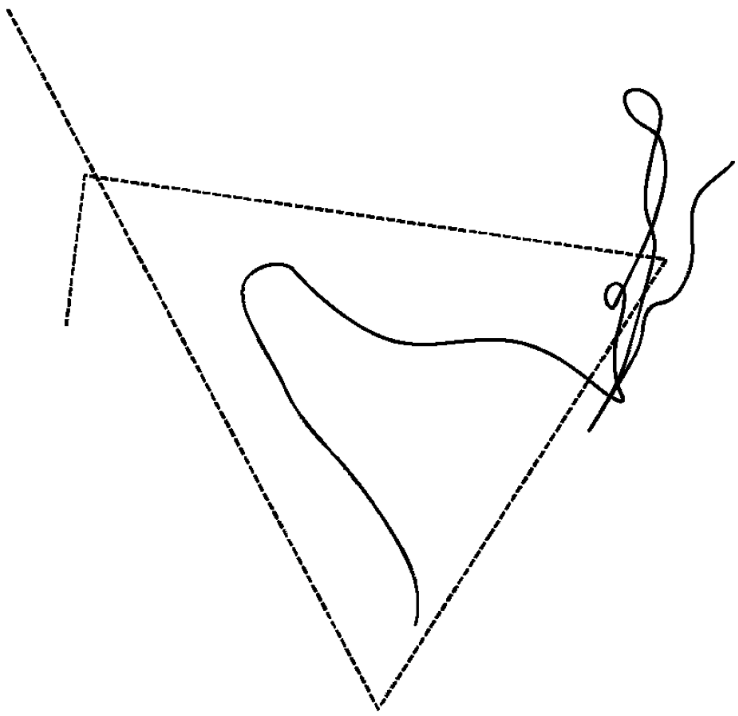


Figure 14: Example trajectory of a node moving with SIMPS (solid line) and one running the RWP (dotted line) (From [BLDAF09]).

thermore, it has not been shown if the model possesses a stationary regime. Indeed, as the behavior of each node can be affected by the behavior of all others, the whole system is in constant evolution, and some simulations may end up to include untypical patterns. Finally, because of the potential dependency between all nodes, and the sharp behavior changes (there is no smooth transition from socialization to isolation, or vice versa) the model may be difficult to be approached for a more complete mathematical analysis.

3.1.2.3 *SLAW*

Node movement generation in Self-similar Least-Action Walk (SLAW) [LHK+09], Fig. 15, is divided into days and consists of several distinct stages. Most importantly, SLAW successfully models several real-world walker movement characteristics. Here, we give these characteristics, along with the part of the model, which produces each of them. A detailed description of the different stages of the model are given afterwards.

1. Truncated power-law flights and pause-times. The pause-times are explicitly generated to follow a truncated power-law distribution. The distribution of the flights, however, comes from the usage of the *Least action trip planning* algorithm and the fact that the gaps between fractal points follow a power-law distribution.
2. Heterogeneously bounded mobility areas. Resulting from the limited subset of clusters and waypoints from these clusters visited by each node and the fact that all nodes have different subsets.

3. Truncated power-law inter-contact times. Produced by the joint characteristics of all stages.

4. Fractal waypoints. Explicitly generated.

On the first stage the waypoint generator produces a given number of points which are going to be used as destination points by the nodes. In SLAW, a fractal point generator is used, following the findings of the same authors which studied in [LHK+08] 5 sets of pedestrian GPS traces and discovered that the waypoints of human walks may be modeled by fractal points. Afterwards, the points are grouped in clusters, by using a simple disk model with radius 100 m (that is, if two points are less than 100 m away, they are in the same cluster).

On the second stage, each node is attributed a fixed set of waypoints to be visited by the node every day. First, 3 to 5 clusters are chosen, each cluster being randomly drawn with probability proportional to its size, i.e. a cluster is selected with probability c/T where c is the number of waypoints in the cluster and T is the total number of waypoints. Then, 5% to 10% of the waypoints of these clusters are selected as the waypoints to be visited every day by the node, independently of the other nodes. Finally, a starting point is chosen with uniform probability among the waypoints assigned to the node.

Each simulation day lasts 12 h. In order to introduce a random element in the node movement, in the beginning of the day new waypoints are added to the fixed set generated in the previous stage. The new destinations are drawn from a cluster different than the clusters selected in the previous stage, which is chosen with uniform distribution, i.e. if all clusters are denoted with C, and the set of clusters chosen for the given node is denoted with C_{node}, then the new cluster is selected with uniform distribution from the set $C - C_{node}$. As with the other clusters, only 5% to 10% of the waypoints are chosen.

Once the complete list of waypoints to be visited during the day is generated (e.g. the list of fixed waypoints plus the list of random waypoints), the node visits them one by one without repetition, moving at a constant speed of $1m/s$ in straight line between the waypoints. At each waypoint, the node pauses, with the pause times distributed according to a truncated power-law. The node starts from its starting point and end the day in the same point, with the pause times being adjusted in such way, as for the entire movement to take 12h.

The order in which the waypoints are visited is determined by a probabilistic algorithm called *Least action trip planning*. In essence, this algorithm randomly selects the next waypoint to be visited, with the probability of each waypoint diminishing with the distance toward it. If the distance between the node and a given waypoint is d, then the weight if that waypoint will be $1/d^{\alpha}$ where α is a parameter. The probability of each waypoint to be selected is its weight divided by the sum of all waypoint weights. Once a waypoint is visited, it is removed from the list of destinations, the probabilities for all waypoints for this node are recalculated, and the procedure repeats.

Any of these stages may be replaced with a different process, which generates the same type of output. The authors have tried using waypoints extracted from GPS traces instead of the fractal point generation process and obtained almost identical results. They also proved, that the length of the gaps between the waypoints follows a power-law

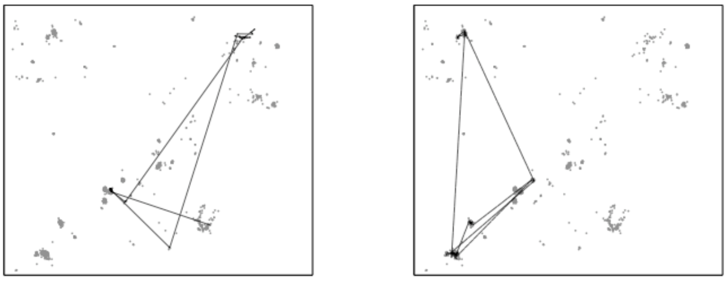


Figure 15: (left) Real GPS trace of a single user. (right) SLAW generated trajectory for a single user. (From [LHK+09]).

distribution when the waypoints are generated with a fractal process. This observation matches the real-world traces they analyzed.

The model exhibits realistic behavior according to several very important metrics, mentioned earlier. However, these characteristics are tested separately, and it is not clear if the joint distributions are going to behave as expected (e.g. the pause time distribution does not depend on the visited waypoint, which means that although a node will pass every day via that point, it will have significantly different behavior). Furthermore, the waypoint selection algorithm choses each waypoint only once per day, which is not the case in many everyday situations.

3.1.2.4 *Community-based*

The mobility model introduced in [MM06] is based on the principles of social interactions and uses findings coming from various studies of real-world social networks such as the network of scientific coauthorships.

In the model nodes are organized in a community, which is represented with the help of a weighted, undirected graph, where vertices correspond to nodes, and edges - to node relations. The weight of an edge denotes the degree of interaction between its vertices (nodes), and may vary between no interaction (0) and maximal interaction (1). The interaction degree determines the geographic proximity of two nodes and depends on the time of the day and the day of the week, i.e. it is not an abstract friendship relation which remains stable during the day. As an example, in the interaction graph during a workday the relations between coworkers would be very strong, while family members would be the dominant interaction partners during the evenings.

The social interactions graph is generated in two steps. First, a scale-free, clustered graph is generated with the Caveman model [Wat03] which starts with K distinct fully connected graphs (representing isolated communities) and one by one rewires the edges with probability p to a node of a different community. Then the nodes are divided into groups based on the formed clusters, each node belonging to exactly one group. Unfortunately, the authors didn't mention how they generated the exact interaction weights once the groups were constituted, so we will assume that whenever an edge exists in the generated graph from step one the interaction is 1, and 0 otherwise.

The simulation area is divided into squares, and each group is assigned a "home" square. All nodes are associated and then randomly dispersed to their respective home squares. The movement consists in first selecting a new square to be associated with (potentially the current one), randomly choosing a point in it, and then moving with constant speed drawn uniformly in $[1m/s; 6m/s]$. If the same square is selected, this model acts like RWP restricted to it.

For a given node, the next association square is selected by looking which is the most attractive square. The attractiveness of a square for a particular node depends on the other nodes going to that square (i.e. associated with this square). More formally, the attractiveness of a square for node i is $\sum_{j \in C} m(i,j)/|C|$, where C is the set of all nodes associated (going) to that square in the same moment and $m(i,j)$ is the weight of the edge (i,j) (i.e. the interaction between them). The square with maximal attractiveness is always chosen.

The simulations are performed for a given day, and every 8 hours a new social interaction graph is generated, e.g. all communities are reinitialized and node interactions are completely regenerated. This is akin to simulating 8 hours of virtual time, then saving the positions of the nodes and using them for a new simulation, repeating this several times.

The model successfully produces power-law distributed contact durations and inter-contact times, with the exponent mainly dependent on the probability p with which edges are rewired during the social interactions graph generation. However, it does not seem to show a pronounced exponential tail which is also considered characteristic for these two metrics. As it is the case with SIMPS, no mathematical analysis of the properties was performed, and it is not clear if there is a stationary regime, and more importantly, if this regime is useful, e.g. if in the stationary regime all groups are isolated each in its own square the model would turn into a scenario of RWP (which will most probably depend on the probability p).

3.1.2.5 *Other*

We are going to discuss two models - *Pragma* [BDdAF05] and the Clustered Mobility Model (CMM) [LYD06]- together, as both adapt the principle of preferential attachment used normally for the generation of growing scale-free graphs. Both models use the notion of attractors, which attract nodes proportionally to the number of nodes staying or going towards the individual attractor. In Pragma the attractors are points, which appear and disappear following a Poisson process, while in CMM the attractors are static zones. Both models provide heterogenous node distribution. *Pragma* manages to create power-law popularity of the attractors. CMM has been studied much more throughly, but it generates exponentially distributed inter-contact times and contact durations [LHK+09], which is not considered realistic.

Another interesting model is the Ant-colony model [LTYY04], where node movement is inspired by ant colonies. In the model, food depots are created on the simulation area, ants having to transport the food to the nest. Initially, an ant moves randomly on the simulation area (presumably with RW), until it reaches a food source. At that point, it takes some of the food, and goes back to the nest, leaving on its way home a mark (pheromone). If a free ant crosses the pheromone, it will follow it. The pheromone weakens with the time. The model has not been analyzed in any way, but its approach is curious.

3.2 EMPIRICAL AND DATA-DRIVEN MOBILITY MODELS

Trace-based mobility models were considered by some as more realistic than the synthetic ones. This, however, depends on the type of traces. Coarse-grained traces provide only a rough view of user mobility patterns, with some important aspects of the mobility left to be deduced by other means. Fine-grained traces (such as GPS traces) include a much more detailed picture of movements, but even they contain small deviations from the real trajectories because of issues with the accuracy caused by deliberately introduced errors, or signal problems caused by surrounding buildings.

Thus, we need to process the traces and model the movements in order to obtain a usable microscopic patterns. Furthermore, modeling the traces gives the possibility to generate multiple scenarios.

Trace-based models have rarely been analyzed, even though most of them are based on usual stochastic models.

3.2.1 *Coarse-grained Trace Based Models*

These types of data capture only medium- to big-scale movement patterns. For example, the error in the position of a device associated to an AP can reach up to 400m. Additionally, trace gathering mechanisms often introduce errors in the time domain - SNMP polled association patterns have a granularity of 5-10 min, and terminal switch-off is detected after half an hour of inactivity.

Data treatment is often the central part of the proposals, and most models fail to provide adequate validation. Furthermore, model definition is very frequently mixed with parameter extraction making it difficult to use the model with another set of traces (one would typically have to posses the same set of data, which is rarely the case). The last point is true, even though the models themselves are often quite simple. Additionally, the fitness of the selected models when compared to other alternatives is never investigated. Why was a second-order Markov chain selected? Does it produce better results than a simple probability distribution? If a new set of traces should be modeled, which approach should be selected? These are questions that remain unanswered, but directly affect the applicability of the proposals.

3.2.1.1 *WLAN trace based*

3.2.1.1.1 KIM ET *al.* [KKK06]

The model proposed in [KKK06] is based on characteristics extracted from WLAN association traces.

3.2.1.1.1.1 Data description The authors studied 1 year of WLAN association syslog traces recorded at the Dartmouth university. They analyzed only the movements of 198 VoIP clients, because typical laptop users have a nomadic connectivity behavior and less can be said about their continuous mobility patterns. The GPS positions of all APs were used later on for the model definition.

3.2.1.1.1.2 Data preparation Only data in the period 8:00h-18:00h were treated, in order to avoid forgotten devices to bias the results.

Then, the traces were divided into a mobile and a static set, by looking at the User daily diameter (as defined in 3). If the diameter is less than 100 m, then that user's trace for the day is declared static, mobile otherwise. The traces are divided into walks whenever the device was switched off for more than 30 min. This way the 3252 mobile traces were converted to 3838 walks, and the 3876 stationary traces into 4006 walks.

3.2.1.1.1.3 Mobility model There are two types of mobility modeled in this paper - stationary and mobile nodes. The simulation area contains n zones, which are used to model hotspot regions.

Stationary nodes are modeled with the help of the following parameters:

- Initial zone distribution.
- Start time distribution.
- Pause time distribution.

A static node enters the simulation at a time drawn from the *Start time distribution*. The hotspot location at which it is introduced is drawn from the *Initial zone distribution*. Finally, the lifetime of the node is drawn from the *Pause time distribution*.

Mobile nodes' parameters include:

- $(n+2) \times (n+2)$ zone transition matrix.
- $(n+1) \times (n+1)$ waypoint count matrix.
- Speed distribution.
- Start time distribution.
- Initial zone distribution.
- Pause time distribution per zone.

A mobile node is introduced to the simulation at a time drawn from the *Start time distribution* in a zone drawn from the *Initial zone distribution*. The next zone to be visited is then drawn according to the *zone transition matrix*, where in addition to the zones in the simulation area are included the probability of moving to the region outside the zones, and the probability of switching the device "OFF". Once the destination waypoint is selected, the number of intermediate waypoints to be visited by the node is drawn from the *waypoint count matrix*. The intermediate waypoints are then generated by uniformly dispersing points in a rectangle whose diagonal is given by the start point and the destination point. These points are then visited in an order relative to their distance from the start point. The node moves with a constant speed, which is drawn from the *Speed distribution*. Finally, once arrived at the destination, the node pauses for a duration randomly chosen with distribution taken from the *Pause time distribution per zone*. The process repeats until the end of the simulation.

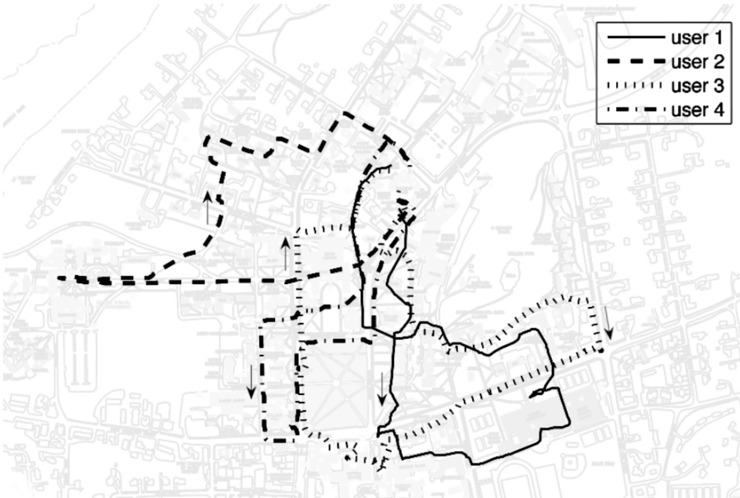


Figure 16: GPS tracks of the controlled walks on the Dartmouth campus map. (From [KKK06]).

3.2.1.1.1.4 Parameter extraction The authors investigated several ways of estimating the exact user trajectory from the list of AP associations. Four methods were used - directly linking AP's centers; using three successive association points to define a triangle, and then taking its medicenter; taking the centroid of all APs which were visited in the past q seconds, repeated every p seconds; employing a Kalman filter to smoothen the data (Fig. 17).

In order to calibrate the Kalman filter, and to estimate the accuracy of all methods, the authors performed four controlled walks (Fig. 16). During these walks, the users recorded their positions with a GPS-device, while in the same time carrying a Cisco and a Vocera device. The Vocera devices associate much more aggressively to new APs when compared to Cisco phones, which tend to remain connected to the same AP as long as possible. This was reflected in the accuracy of user position estimation - all methods performed considerably better when used on Vocera walks. However, apart from the AP center connection method (which was used as a worst-case comparison), the other three methods did not show considerable differences in the trajectory estimation. The authors preferred using the Kalman filter technique for the set of mobile traces, and the triangle centroid for the static traces.

User pause times were extracted by assuming that in most cases people are walking across the campus, and thus move with speed in a well defined range. The average speed of a user is defined as a moving average bootstrapping the process with a typical walking speed value of 1.34 m/s. For each trace segment the movement is divided into pause in the beginning and move period with user's average speed. The euclidean distance between segment's start and end points is divided by the average speed to determine the moving time, and the rest is considered as the pause time. The distribution of the speeds is obtained by using the segment average speeds weighted by the respective segment lengths. In the end, the authors discovered that the average speed and the pause times follow log-normal distributions.

Hotspot locations were extracted from the movement traces by overlaying a 2D Gaussian distribution with $\sigma = 20m$ at each pause location.

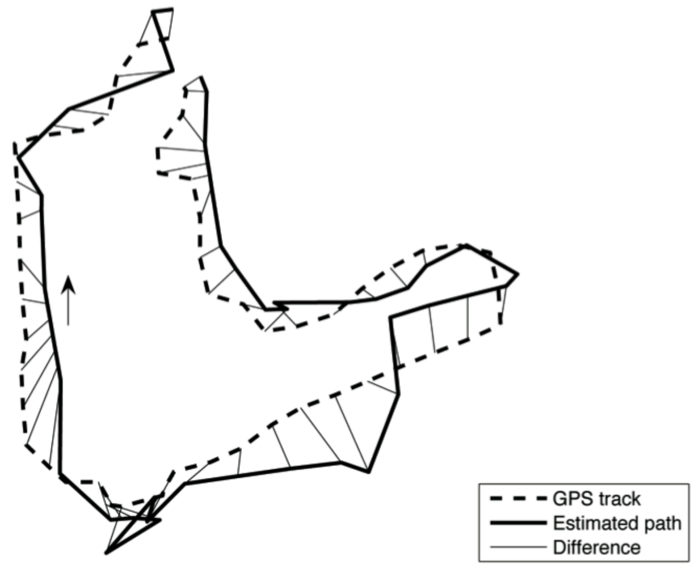


Figure 17: Differences between the GPS track and the path estimated using a Kalman filter. (From [KKK06]).

The distribution is weighted with the pause duration. Afterwards, a threshold value is selected, with the regions having more than that value being declared as the hotspots. The various hotspot-related distributions required by the mobility model are calculated.

Initial node and start time distributions are based on the first AP and time of day at which a node has associated to the network.

3.2.1.1.1.5 Model validation The number of nodes per region per hour has been measured, and the values produced by the synthetic traces and the real traces were compared. The relative error for each hotspot is determined by the formula $\sum_{i=1}^{n} |r_i - s_i| / \sum_{i=1}^{n} r_i$, where r_i is the number of real users during the i-th hour, and s_i is the simulated users in the same hour.

The median error for the five regions in the simulation is 17%. One of the hotspots, however, has a significantly higher error of 46% (Fig. 18).

3.2.1.1.1.6 Discussion The paper introduces various approaches to WLAN trace analysis and modeling. Unfortunately, as with other WLAN trace-based models, it does not provide sufficient validation of the results. Furthermore, in spite of the multitude of aspects extracted directly from the traces, the error rate is high, for one of the five hotspot even reaching 46%. The authors have not pointed out how was the ping-pong effect handled, yet it can affect the results by increasing the rate of mobility in the network under study.

The trajectory-estimation algorithms were evaluated with a very restrained set of control walks (only four trajectories of 20 min of walks were recorded) which seems insufficient for statistically significant results. Also, the GPS-traces used as basis for the study contain natural inaccuracies, which are of the order of the differences being studied. Moreover, because of their limitation, the GPS traces were only used for

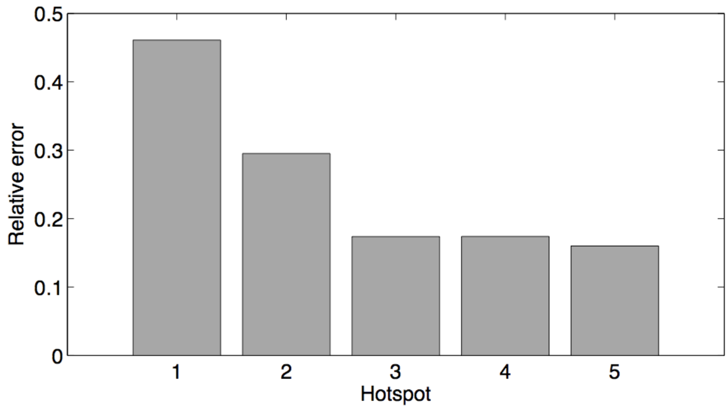


Figure 18: Relative error between the synthetic tracks and the real tracks for each hotspot. (From [KKK06]).

outdoor walks, which means that the trajectory-estimation algorithms may provide significantly different results when used indoors, where the signal propagation and AP concentration may be considerably different.

Finally, the log-normal distribution of the average speed is a surprising discovery, which seems to contradict the widely-accepted model of normally distributed average speed. The question needs further investigation, with eventual corrections to the speed-detection algorithm.

All these points are a direct result of the big number of assumptions necessary to transform such a coarse-grained type of traces as WLAN associations into micro-level mobility.

3.2.1.2 *Activity-based*

The activity-based models are founded on the principle, that a trip is a derived demand, being a means to an end, rather than an end of itself [SK99a]. An activity is a collection of actions typical for the considered scenario [BKOKR04].

Here, we are providing the way activity modeling was implemented in [SK99a,SK99b], but it was also used as a foundation of other mobility models, e.g. [Ste02, Ray03, BKOKR04, CB05, KB05, ZHL06, KRKB06]. However, even if there are some differences in the way activity is defined, or the accompanying characteristics implemented, the core of the approach remains unchanged. A survey of the activity modeling techniques may be found in [AG92].

3.2.1.2.1 DATA DESCRIPTION

The activity transition and duration matrices used by this mobility model were derived from a trip survey [Ass89] where a travel diary was completed by each household member over 5 years of age, in which details on all trips taken during the survey day were recorded. Each recorded trip included the trip start and end times, the trip purpose at the origin and destination, and employment status.

Trips are classified in nine categories: work, work-related, school, serve passenger, shopping, social/recreation, personal business, return home and other. The day is divided into twelve equal time periods which are used to aggregate the data from the survey. The authors observed that there exist several types of behavior and decided to define four categories of persons:

1. Full-time employed outside the house
2. Part-time employed outside the house, but not student
3. Student, secondary or post-secondary, possibly employed part-time outside the home
4. Not employed outside the home, and not a student

Each category possesses its own parameter values. Hereafter, the description of the parameters and the algorithms are given for a single user set. The existence of several user sets can be achieved by running simultaneously several such "simplified" models, which differ in their userset-dependent parameters.

3.2.1.2.2 MOBILITY MODEL

The following parameters are used for mobility generation:

- Mean speed.
- Activity transition matrix.
- Activity duration matrix.
- Per-activity zone weight.
- List of intermediate zones to be visited when moving between two zones.

The simulation area is a 2D rectangle, divided into equaly-sized zones. A zone may have weight associated to it for each of the activities, although in the article only zones with shopping activity have associated weights (representing the zonal retail employment). An activity is modeled as the triplet $(\mathtt{timeofday}, \mathtt{duration}, \mathtt{zone})$. In the beginning of the simulation, each node is assigned a current activity (e.g. home) and typical zones for some of its activities (e.g. home, school and work zones).

The model is then executed by first selecting the next activity to be carried out A and its duration. The activity is drawn randomly from the activity transition matrix, and depends on the current activity and time. The duration T_A of this activity is chosen from a distribution depending on the next activity A and the current time. Then, if there is a typical zone for the activity, it is selected as the destination zone. Else, if there are zones with weight associated for this activity, the destination zone is selected randomly from the top five, after they were ordered in a descending order on their weight divided by the distance to them. Otherwise, the destination zone is randomly selected. Once the destination zone is selected, a trip is created from the list giving the zones to be visited when going from the current to the destination zone.

3.2.1.2.3 DISCUSSION

The approach of activity-modeling provides many opportunities to realistic mobility modeling. However, it requires a detailed activity list for a large population - data which is particularly difficult to obtain. Additionally, it may only be used to model existing human societies, which means that making predictions for future scenarios, or simulating other types of entities (e.g. animals) is not going to be straightforward.

However, if activity data is available, it may provide realistic macro-level behavior. Additionally, based on the activities, one may infer other types of information regarding the simulation scenario which could lead to more realistic simulations as a whole. For example, the simulation creator may assign different classes of data traffic to the different activities.

A drawback of this approach is that in general it requires a lot of background information and a lengthy algorithm description in order to detail the simulation scenario. For that reason most of the activity-based mobility models were not validated, or validated very briefly. The lack of such validation, together with the complexity of the parameters have a serious negative impact on the acceptance of such models.

3.2.2 *Fine-grained Trace Based Models*

There exist several technologies which may be used to continuously monitor the position of a given subject with high accuracy (< 10m error). The most prominent one is using GPS tracking. Even though the performance is degraded in indoor environments, the ease of use and its availability make it a perfect choice for such studies. The alternatives include analyzing and extracting the position from video recordings or WiFi/cellular network triangulation [Var06, SGS+06].

3.2.2.1 *GPS trace-based*

There have been numerous studies in the traffic engineering community which are based on or analyze the usage of probe vehicles as ways of real-time traffic estimation [HZW+07], Origin-Destination (OD)-matrices extraction [EL04], arterial speed estimation [ZXZ05], road selection [KYYM03, MM03, LLN06], trip duration estimation [MSM04], the effects of rain [WYMM06] and drainage-improved pavements on the traffic [THM+04], etc. However, to date, there exist only few mobility models which include some form of GPS-trace foundations.

One of the problems with these traces, is that they are difficult to obtain. Indeed, because of the cost of the individual GPS devices, organizing a medium- scale study (100+ participants) is too expensive, and obtaining traces gathered by commercial fleets (such as trucks or taxis) is difficult as the companies are reluctant to expose potentially vital business information.

Recently, there have appeared some works that use GPS-traces to produce micro-level mobility. The first is the synthetic HRW (presented earlier) which exhibits characteristics discovered in freely available taxi- and volunteer- GPS traces [PSDG08, PSDG09b]. Lévy walks [LHK+08, RSH+08] have been shown to realistically model human displacements by using GPS-traces generated by volunteers. The same principles were used for *SLAW* [LHK+09], as presented earlier. Another interesting work is the trace-based analysis [HLL+07] which studies the performance of several DTN protocols by replaying the traces gathered by 4000 taxis in Shanghai, China.

In [SMR05] the authors proposed a possible mechanism of inferring activity information from GPS-traces, but without validation. Extracting significant locations from GPS traces and predicting the next one to be visited is discussed in [AS03].

Overall we can conclude, that the huge majority of GPS-trace studies lies outside the range of mobility modeling for wireless networks. They provide indispensable information about the fundamental characteristics of different types of movements, but their complete usefulness to the community is yet to be harnessed.

3.2.3 *Map Based Models*

Defining a realistic model also requires carefully modeling the node environment, an essential part of which are node movement constraints. One of the most frequently used constraints is limiting movements to a graph embedding (as described in Section 3.1.1.4). Extracting this graph from real-world maps is the base component of several mobility models, both aimed at pedestrian and vehicular movement simulations. Here, we will present some of these models incorporating the typical approaches seen in the literature.

3.2.3.1 *Pedestrian*

Some of the models we introduce in the appendix simulate pedestrian movement based on graphs modeled after real-world maps (see A.3.1.1.2 and A.3.1.3).

Another example of such model is the Urban Pedestrian Flow [MSK+05]. The edges of the graph of this model represent streets, while the vertices stand for points of interest (such as stations, shopping centers, ...) or street junction elements (intersections, entrances, etc.). The graph is weighted, where the weight of a street denotes the width of the street segment. Nodes are produced or destroyed at a fixed set of vertices, denoted with V'. Pedestrians are divided in groups, each group having a specific route pattern called a path. A path is modeled as an acyclic sequence of vertices, with both ends part of V', with shortest path movement between two vertices.

Pedestrian flow rate on paths are then estimated based on observed pedestrian flow rates on the edges (street segments) of this path. The authors used linear programming technique to deduce the pedestrian flow rates, by minimizing the maximal error between observed and deduced path flow rates. The model is validated with the help of pedestrian movements extracted from video-recordings of several street segments from Osaka, Japan (Fig. 19). The density of each street was modeled, and the calculated path flow rates were compared to the real path flow rates.

After tuning the objective function to be optimized, the maximum error dropped from 35% to 8.67%. The latter value indicates that this approach can faithfully estimate the path flow rates given that edge flow rates are available. This approach is based on traffic flow estimation with pedestrian count, which has already been widely used in the traffic engineering literature (e.g. [vZB82]). The main problem is that

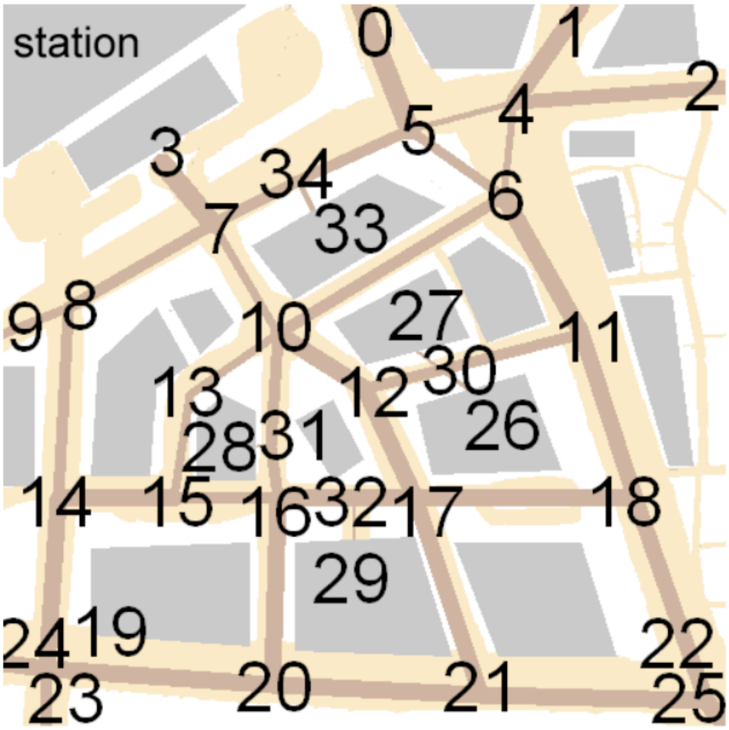


Figure 19: Map of the area used for validation of the Urban Pedestrian Flow model. (From [MSK+05]).

it requires a considerable amount of observations in order to provide faithful results, and that number grows very fast with the size of the area to be simulated. Fine-tuning the model requires detailed domain knowledge, or the observed errors could be very significant.

3.2.3.2 *Vehicular*

Apart from the synthetic models using artificially specified maps discussed in Sections 3.1.1.4.2, A.2.1.3.1, and A.2.1.3.2, almost all VANET simulations use graph constraints extracted from real-world maps (Fig. 20).

The different models vary in their complexity, with the most simple performing random walks on the graph, and the most complex approaching the characteristics of complete city traffic microsimulations [NBG06, BHM07, FHFB07].

The proposed solutions include modifying the intersection handing behavior (for example stopping at every intersection, or depending on the speed traffic [MPGW05, PM06]), addition of traffic lights regulating the car flows [FHFB07], car following behavior [Ste02], intersection turn probabilities (e.g. proportional to the width of the streets or geared towards a given hotspot [NKK02]), path selection [MHMA06] and so forth.

Graphs are most frequently extracted from standard Geographic Information System (GIS) file formats (such as MapInfo [NKK02], Economic and Social Research Institute (ESRI) shapefiles [SJ04], and Geographic Data Files (GDF) files [FHFB07, Ste02]). A prominent depot of freely available map data is the TIGER database compiled by the U.S. Census Bureau [Cen].

Almost all VANET simulators provide the possibility to choose between OD-matrices or activity modeling in order to discover the high-level movement patterns of the users. Then, trip selection is also configurable, with shortest N-paths as the most popular default choice. Car following behavior also can be configured, with the Intelligent Driver [THH00] being considered as a realistic choice.

Additionally, because typically VANET simulations require interaction between the network and the mobility simulators, some of the VANET mobility simulators come prepackaged with their own network simulator (e.g. [WC09, MWR+06]), meaning that the entire simulator package must be validated. Another possible approach is to interface an existing traffic simulator with an already affirmed network simulator, as done in [PRL+08]. Finally, there exist more classical simulators which produce traces that are fed to the network simulator [SHH02, KML07, NBG06, FHFB07] (e.g. there is no simulator cooperation).

However, with the increase of the involved microscopic movement characteristics, the models become more difficult to use and configure. The models start converging towards the traditional traffic microsimulators (such as VISSIM [Fel94] and TRANSIMS [NNB+99]) and as a result have the same drawbacks in the context of wireless network simulations. Worst of all, the outputs of most of the detailed VANET models are never really validated, considering that conceptual model validity is sufficient. However, there are many facets of node movements and basic sanity checks should be performed. For example traffic microsimulators have been validated against characteristics typically studied by road traffic engineers, such as congestion formation, bottlenecks caused by lane merging, etc. This has not been performed on specialized VANET simulators, with the notable exception of [FHFB07] where the authors have shown that the model produces some realistic phenomena, such as speed and density shock waves. In addition, even the dedicated traffic microsimulators do not guarantee realistic behavior and may produce significantly different results when compared to each other [CMS03].

Additionally, none of the simulators were analyzed with network-relevant metrics such as ICT or Contact Duration (CD) distribution. Instead, usually a comparison between several MANET routing protocols is performed, with specific packet metrics reported. Several simulators with intuitive graphic interfaces have been proposed in the recent years [MWR+06, KML07, WC09], yet no parameter exploration has been made, and no indications on how to obtain realistic movements are given, and as shown in [FHFB07] obtaining realistic behavior requires careful selection of the used submodels.

Finally, the core of all-encompassing detailed microsimulation is to obtain realistic behavior of the entire community by simulating thoroughly the individual behaviors. Yet, the VANET models have not yet reached the level of sophistication of the true traffic microsimulators, which may lead to potentially unrealistic movement patterns even when they are perfectly configured.

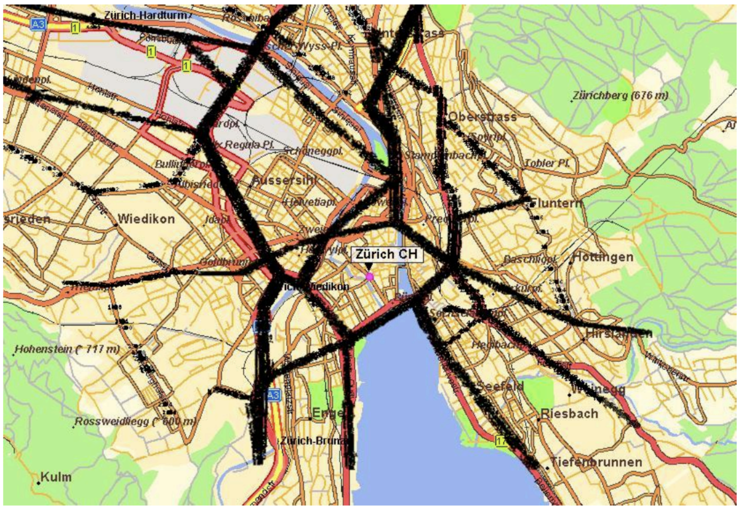


Figure 20: Superposition of generated mobility traces and a map of the center of Zurich. (From [NBG06]).

3.3 additional trace analyses

There are many additional studies of various user-generated traces that are relevant to human mobility, albeit not providing an exact microscopic mobility model definition. Example of these traces include university campus schedule [SMO06], GSM [EP09], Bluetooth [CHD+07], WLAN [HH05], e-mail [SCJ04] and GPS traces [KLBV07].

In this section we will overview the major studies dedicated to WLAN, Bluetooth and GSM trace analyses. WLAN, GSM and GPS traces log the absolute location of the devices (although with a possibly large error). On the other hand, Bluetooth traces register only the relative information (i.e. when two nodes are close to each other). For that reason, Bluetooth traces are typically used for node encounter studies. Encounter traces may be obtained from absolute position data under certain assumptions (e.g. two nodes are in contact if they are associated to the same AP in the same moment). Even though there are works that have addressed the inverse process (generating absolute positions, or at least partial spatial information from the encounter traces [Wal08]) it remains largely an opened question whether if there exist an algorithm to consistently reconstruct approximate absolute locations from real-world encounter traces.

3.3.1 *WLAN Association Traces*

Whenever a device is connected to a WiFi network, certain amount of information can be gathered, such as the current Access Point (AP) to which it is associated, when and how it authenticated (if there is authentication in the network), if it roams across the different APs, the traffic counters, and so forth. We are going to be interested only in data related to device localization, i.e. we are not going to discuss any traffic related analyses. An important remark about WiFi networks is that their deployments reflect the typical usage patterns and as such are not optimized for providing a complete coverage of their environment, e.g. in a typical campus the aim would be to enable classrooms and residential buildings with high-speed, reliable WiFi access, while the green spaces and the parks are going to be addressed only at a later stage. This is one of the reasons why mobility information is so scarce in these traces.

User association information is obtained directly by interrogating the APs, while authentication logs are gathered from the authenticating servers, e.g. RADIUS, captive portal. There exist two possible mechanisms with which AP association traces can be gathered - via periodic SNMP requests (polling-based logging) or via automatically generated syslog events (event-based logging). In both cases a dedicated server collects the data. For polling-based logging the server consecutively performs SNMP requests to every AP in the network at regular intervals. In the event-based case, the APs are configured to send automatically this information to the logging server, whenever an important event occurs (association, roaming, etc.). Both methods have been extensively used for trace collection and analyses. It has been pointed out that event-based logging is more appropriate for mobility-related characteristic analysis than the polling-based [HH05].

Because association traces contain as only identification user's device MAC address, it became a widely-accepted assumption to define that each MAC address uniquely represents a user. This assumption is not always true, as some users may share their devices, while others may possess more than one. Furthermore, users may change their devices (e.g. upgrade their laptops), in which case they will be counted as completely different persons. In a previous study of a network with authentication [PDN07] we have shown that 19% of all devices were actually used with more than one login identifier. In the same work we proposed a procedure for using a better user identifier - the *primary MAC address* (the MAC address with which a login has been online for most of the time). However, authentication information is not readily available for most of the existing WLAN traces and this procedure is not applicable in these cases. Furthermore, we have performed a small-scale survey among the users of the network we studied in [PDN07] and discovered that whenever a device is shared, its owner is very likely to be close to it, and is frequently also sharing the login (e.g. when a laptop is borrowed, the WiFi authentication password has already been entered by the owner). The results of this survey show that even though the *primary MAC address* provides the best way to identify a physical person, it may give insignificant improvement over a simple device MAC address (unfortunately, we are unable to precisely quantify the difference).

The WLAN trace studies are for the biggest part gathered in university campuses, and for the most of them are freely available from [cra]. However, there exist studies of other kinds of WLAN deployments (e.g. corporate networks [BC03], commercial hotspot networks [BHK05], meetings [LCJ05]) which can be compared to the widely analyzed and studied university networks.

University WLAN traces of the same scale tend to exhibit the same characteristics (e.g. [HKA08,SB04,TP06,HZ02,KK05,PSS05a,PMK06]). Slight differences in some of the observed patterns may be seen whenever there are differences in the device type (e.g. PDA-only trace as the USCD trace in [MV05,HH06b]) or studied area limits (such as the MIT traces limited only to several buildings in [HH06b]).

General finding for any of the studied WLAN traces is the strong time variability of the data - the number of active (associated to the network) users depends on the day of year (Fig. 22), day of week (Fig. 21), and hour of day (Fig. 23). Furthermore, the number depends on the time of the year (e.g. vacation periods have less associations), and has a stable increasing trend, resulting from the increased rate of laptop penetration, evolving WLAN coverage and user adoption of the technology. This includes strong periodicity - both on the scale of APs and on individual users. In [KK05] the authors used Discrete Fourier Transformation and found that the predominant AP usage periodicity is 24h, which allowed them to classify the APs according to their usage peaks in 5 clusters - with morning, noon, afternoon, evening peaks and with no clearly defined peaks. Alternatively, [FERH06, HH06a, jHDH07] focused on the user-centric periodicity (best demonstrated with the *network similarity index* in Fig. 25), and with the help of Principal Component Analysis managed to categorize the different types of users and represent the various AP association patterns with low-dimensional data. In [PSS05a] it was shown that user arrivals at AP may be modeled with a time-varying Poisson process. Another study showed that the association patterns of a conference are highly correlated with meeting starts [LCJ05].

Also, in all university and corporate WLAN analyses, the wide majority of users tend to spend almost all their time in their home location [HH05, HH06b, HKA08, BC03]. The home location is defined as the place (a single AP, or all APs in 50m radius) where the user spends at least 50% of their total active time (Fig. 28). Moreover, a user spends more than 95% off time at its top five most visited APs on average [HH05, HH06b]. This, combined with the finding that only small number of the users are roaming (e.g. 18% for [KE02]) makes this type of traces difficult for direct user mobility extraction. Indeed, users move between places, but because of the inherent difficulty of walking with an opened laptop, in most of the cases people exert nomadic mobility, i.e. move → sit → open laptop and work → close laptop → stand up → move. As discovered later, the ping-pong phenomenon, which appears when a device repeatedly changes its point of association only due to signal fluctuations (no physical movements) is responsible for a sizable fraction of the roaming events in the traces.

Node encounters have also been studied for university WLAN traces [HH05, HH06b], by assuming that two nodes can communicate if they are associated to the same AP in the same time. It turned out, that all nodes encounter less than 40% of the user population within a month and on average, a node encounters 2-6% depending on the campus trace. Inter-node encounter durations follow BiPareto distribution (Fig. 29). Encounters link most of the nodes together in a connected graph, albeit each node encounters only with small portion of the whole population. The encounter graph is a small world graph, and even for short time period its clustering coefficient, average path length, and connectivity are all close to those for longer traces. Because of this, the encounters patterns are rich enough to support information diffusion - information can be delivered to more than 94% of users within two days.

Another prominent characteristic of university campus WLAN is the highly uneven popularity of APs pointed by [SB04] for the University of Saskatchewan, and [KE02] for Dartmouth College. An interesting observation on the evolution of the Dartmouth College's network is that the proportion of heavy users remained unchanged (Fig. 24) [HKA04, HKA08].

An important aspect of WLANs is that session lengths (a session is the duration of uninterrupted user association, e.g. including roaming between APs) follow Pareto (e.g. [BHK05] for hotspot network [BC03] for corporate network, [HZ02] for university network), or BiPareto [PSS05b] distributions. In [PSS05b] the authors found that the session duration is best approximated with BiPareto distribution with a knee at 15 min for sessions that spanned more than one building (Fig. 27) and a knee at 27 min for sessions that included only one AP (Fig. 26). The discrepancy with the other WLANs may come from the fact that in [PSS05b] the authors used more precise, event-based session length estimation, when compared to the polling-based data logging used in [BHK05, BC03, HZ02]. Moreover, AP dwell time (the time a node was associated to a single AP) also seems to follow a Pareto distribution as in the university network studied in [TP06] (note however that the dwell time is different than session duration, because a session continues when a node roams to another AP, while the dwell time is counted only on a per-AP basis). Inter-association time for university networks also follows a power law (e.g. [TP06, HZ02]).

In a study of several university networks [HH05, HH06b] found that the association patterns depend on the type of the used devices. For example, PDAs have a smaller online fraction, generate more associations, visit larger portion of the campus and generate less handoffs than laptops. Additionally, the number of visited APs per user and the session diameter (the distance traveled without interrupting the association) also depend on the type of device [HKA08] - the median number of visited APs is for 17 for laptop users, 9 for PDA users and 61 for VoIP device users, and the median for session diameter is 14m for laptop users, 17.5m for PDA users and 27.8m for VoIP device users. The difference comes from several factors, most important of which is the size and the typical usage of a device. A small wireless VoIP phone which allows carrying out conversations while going to lunch is much more mobility-friendly than a laptop.

To sum up, WLAN studies show that captured characteristics depend on the used device, have strong time variance (depend on the hour of day, day of week, and period of year) and periodicity (with dominating natural periods of 24h and 1 week). Session durations follow Pareto or BiPareto distribution, and inter-association times also follow Pareto distribution. AP usage is highly skewed, with devices typically using up to five distinct APs, and a strong tendency to stay at a single, home AP. An important issue yet to be resolved is a proper ping-pong effect elimination mechanism.

3.3.2 *Bluetooth Traces*

Bluetooth-enabled devices provide one of the most accurate encounter information (e.g. when compared to WLAN or GSM traces). The typical range is 10m and is not limited to a single geographic location (thus, all encounters are logged, which contrasts WLAN traces which are limited to the encounters logged only in the network deployment area).

One of the most prominent traces are gathered with the help of dedicated iMote devices [CHC+05, CHC+06, CHD+07] or mobile phones running appropriate software [EP09]. The analysis showed that the

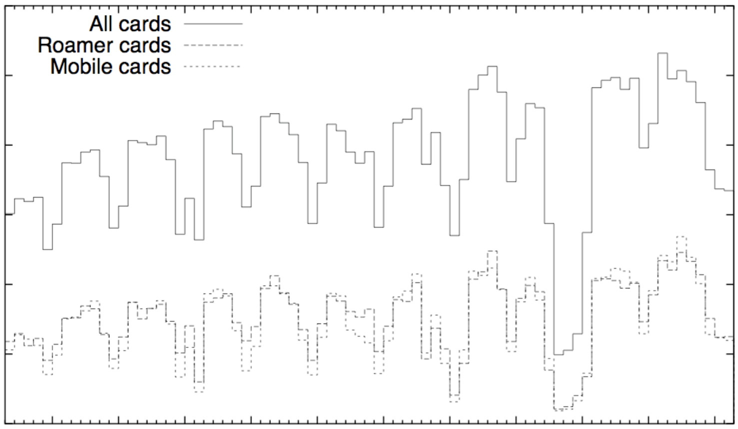


Figure 21: Number of active (associated), mobile, and roaming cards per day. Mobile card is one which associated to more than one AP for the day. Roaming card is a mobile card which roamed from one AP to another. (From [KE02]).

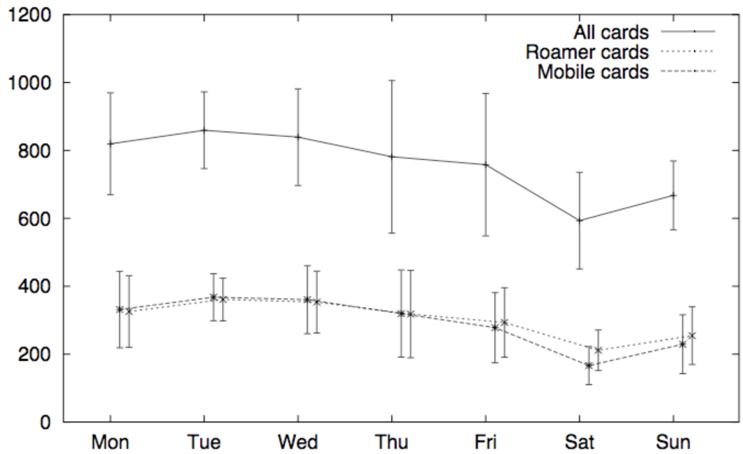


Figure 22: Number of active, mobile, and roaming cards per day of week. The curves show the mean, while the bars show the standard deviation. The three curves are slightly offset, so the bars are distinguishable. (From [KE02]).

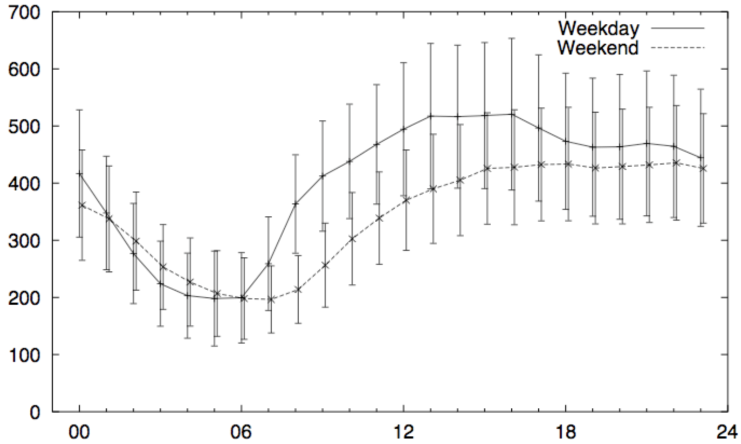


Figure 23: Number of active, mobile, and roaming cards per hour of day. (From [KE02]).

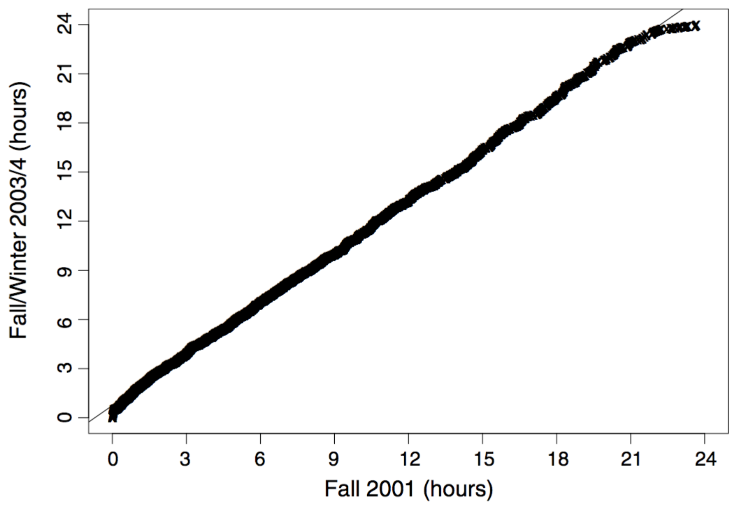


Figure 24: Quantile-quantile plot, average time per day per user. (From [HKA08]).

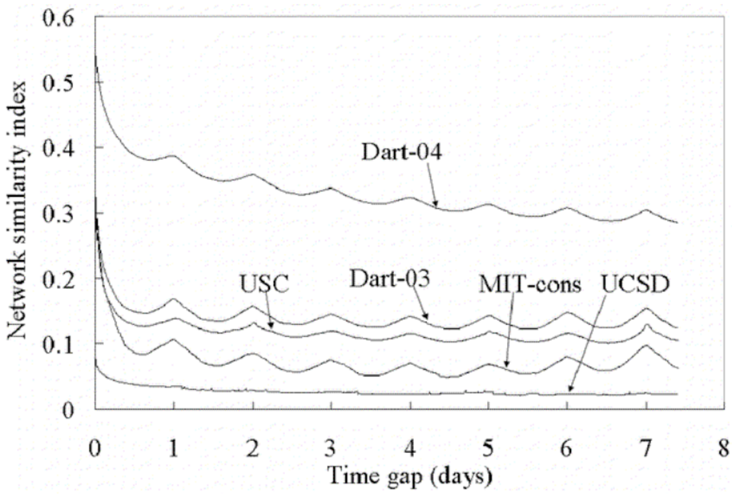


Figure 25: Network similarity indexes. The peaks represent intervals for which there is high location similarity. (From [HH05]).

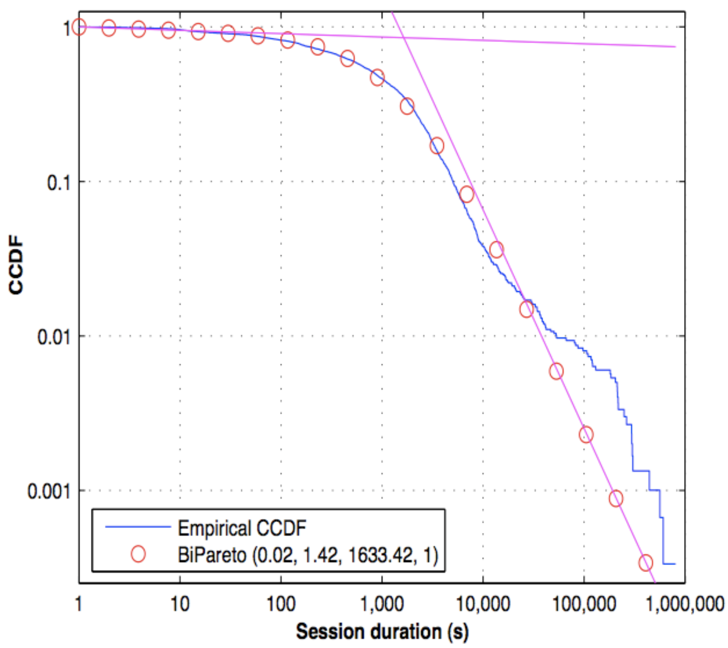


Figure 26: Session duration for single AP sessions: Empirical observations vs model (From [PSS05b]).

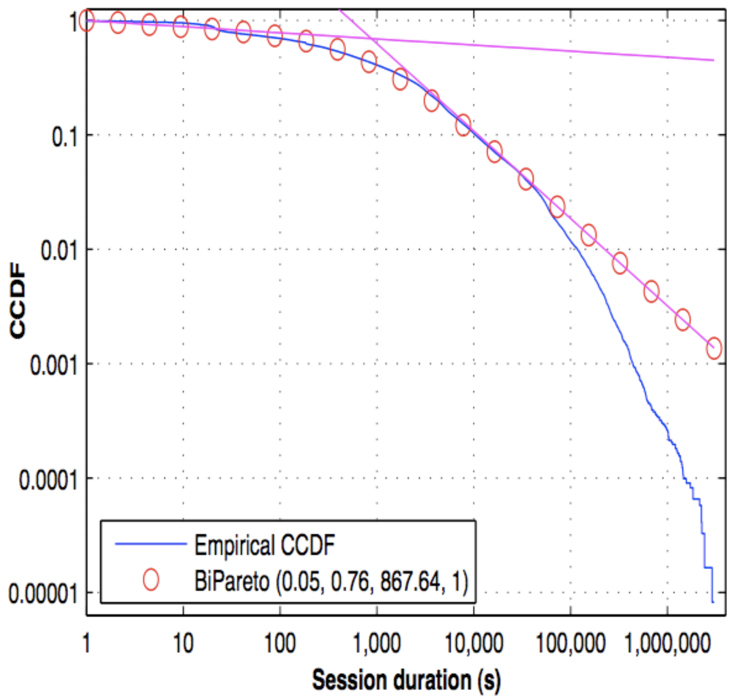


Figure 27: Session duration for sessions with inter-building movements: Empirical observations vs model (From [PSS05b]).

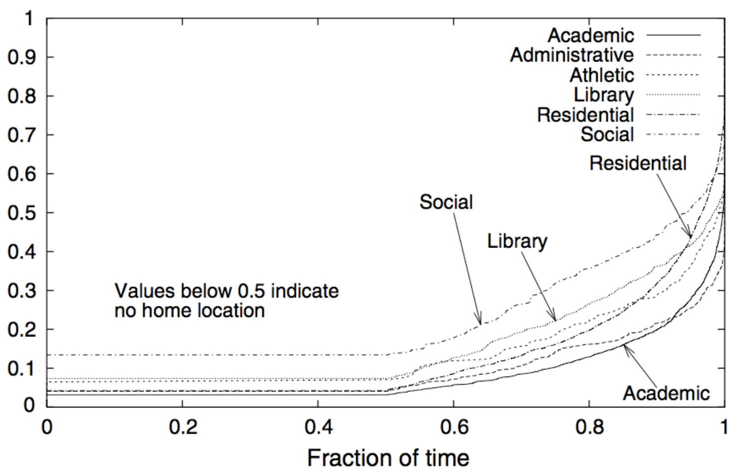


Figure 28: Fraction of time that users spend at their home location, by the building type of their home location. (From [HKA08]).

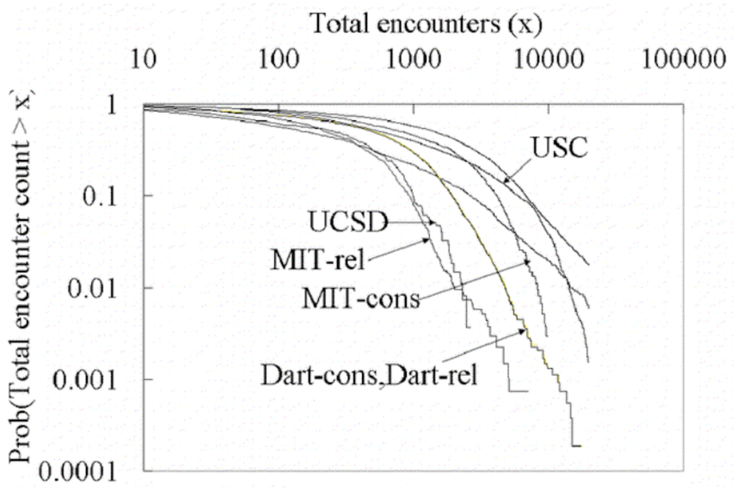


Figure 29: CCDF of total encounter count. (From [HH05]).

aggregated ICT distribution is well approximated by a power law distribution up to a certain limit. A more detailed analysis of the same traces showed that the ICT distributions follow power law distribution with exponentially decaying tail [KLBV07]. The pairwise ICT distributions were also approximated with power law distribution in [CHD+07]. The shape parameters of the different pairs ICT vary significantly, and are in general different than the shape parameter of the aggregated ICT distribution, which testifies to the heterogeneity of the movement patterns. In [CLF07] however, the authors found that even though it is possible to model some of the pairwise ICT distributions with power law distribution, they are better fitted with individual log-normal distributions (although some could also be fitted with power law or even exponential distribution), which combined together produce the aggregated ICT power-law with exponential tail distribution.

In [EP09] the authors analyzed the traces of 100 participants for a period of 113 days. The data were collected with Nokia 6600 smart phones running the Context application from the University of Helsinki [ROPT05]. The information collected included call logs, Bluetooth devices in proximity, cell tower IDs, application usage, and phone status (such as charging and idle).

A profile is built for each user (called eigenbehavior) based on the eigenvectors of the data calculated for the person. The profile of the user is built on the tags *Home*, *Work*, *Else*, *No signal* and *Off* (illustrated in Fig. 30). The day to day activity may then be reconstructed with good accuracy – using the six primary eigenbehaviors gives an accuracy in the range 90% – 96% (see Fig. 31). Each person is attributed an entropy measure, which depends on the regularity of the schedule. Not surprisingly, people have different entropy measures (professors have lower entropy than freshmen students). Furthermore, the eigenbehaviors of the members of a social group tend to be clustered within the same behavior space.

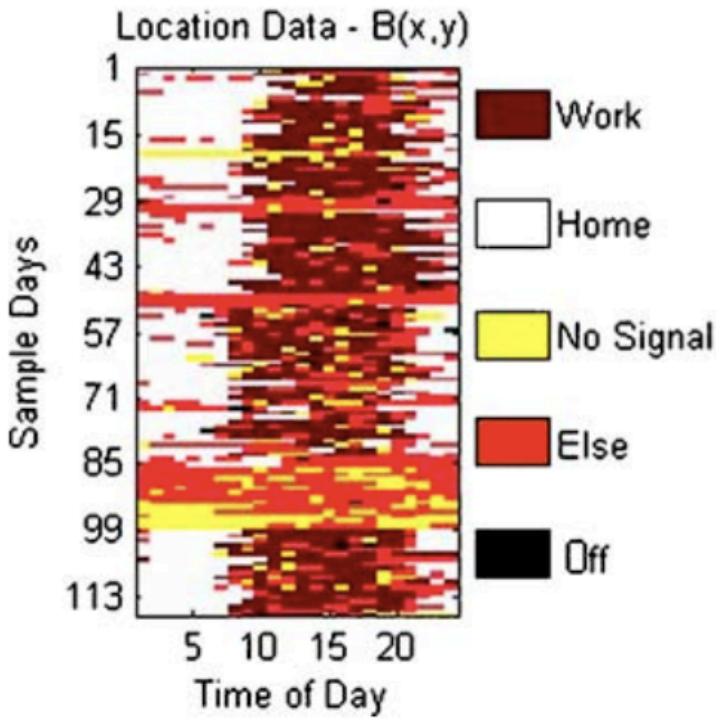


Figure 30: The behavior of a given subject over the course of 113 days for five situations. (From [EP09])

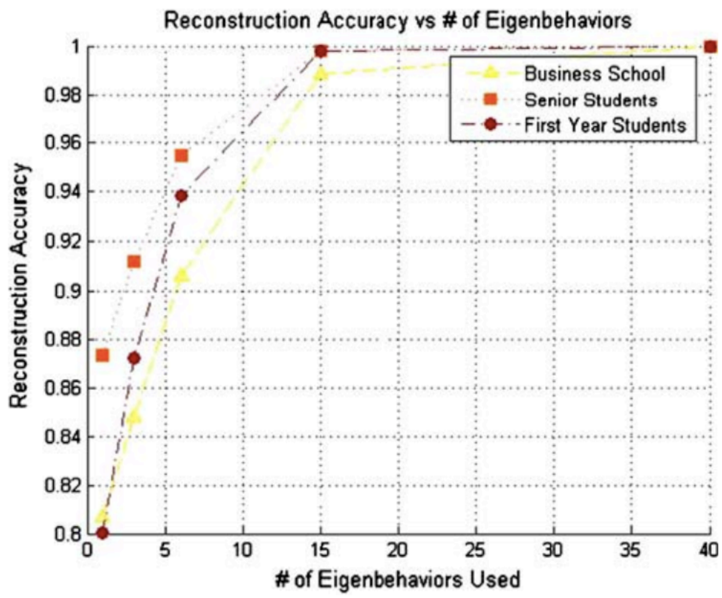


Figure 31: Approximation error (y-axis) for the different subject groups as a function of the number of eigenbehaviors used (x-axis) with the states off and no signal removed (From [EP09])

3.3.3 *GSM Traces*

Mobile phones are almost perfect mobility measurement tools - the penetration rate in some of the countries has surpassed 100% [Wik09b]. A technical limitation of these traces is that they are very coarse-grained - typical cell size may span 1-3 km^2, which completely hides all micro-mobility details. A more serious obstacle to the usage of these traces, however, are the strong privacy issues related to them, as well as the reluctance of mobile operators to unveil such important business information.

Fortunately, a group of researchers managed to obtain such traces from a big european operator, and published significant findings in one of their papers on the subject [GHB08]. The authors studied the movements of 100000 anonymized mobile phone users during a period of six months, and found that the aggregated displacement distribution is well approximated by a truncated power-law:

$$P(\Delta r) = (\Delta r + \Delta r_0)^{-\beta} \exp(-\Delta r / k) \tag{3.1}$$

with exponent $\beta = 1.75 \pm 0.15$ (mean $\pm$ standard deviation), $\Delta r_0 = 1.5\mathrm{km}$ and cutoff values $k_{D1} = 400\mathrm{km}$ and $k_{D2} = 80\mathrm{km}$.

The authors show that in the traces, a population-based heterogeneity coexists with individual Lévy trajectories, thus equation 3.1 represents a convolution of the following hypotheses:

- each individual follows a Lévy trajectory with jump size distribution given by equation 3.1
- the observed distribution captures a population-based heterogeneity, corresponding to the inherent differences between individuals

Further, the users are separated into classes according to their radius of gyration r_g. The radius characterizes the linear size occupied by each user's trajectory up to time t and is calculated as the root mean square distance of the points visited by the user from their center of gravity.

User trajectories are normalized, by using the following procedure [GHB]: The *tensor of intertia* of the user trajectory is symmetric, and can thus be diagonalized. The coordinates of the diagonal tensor are called its *principal axes*. The principal axes of each individual user are then rotated to a common reference frame. Finally, all traces are normalized by dividing the x and y axes by their corresponding standard deviations σ_x and σ_y (see Fig. 32).

After the user traces have been processed in this way, the probability density function $\Phi(x, y)$ that a user is in a given position (x, y) is calculated, and after the rescaling with the standard deviations turns out to be the same for all types of users (Fig. 33). These findings suggests that key statistical characteristics of individual trajectories are largely indistinguishable after rescaling [GHB08].

The findings of the paper suggest that human trajectories exhibit a high degree of regularity of their temporal and spatial statistical properties. Furthermore, each individual is characterized by a significant probability to return to few highly frequented locations and a time-invariant intrinsic travel distance.

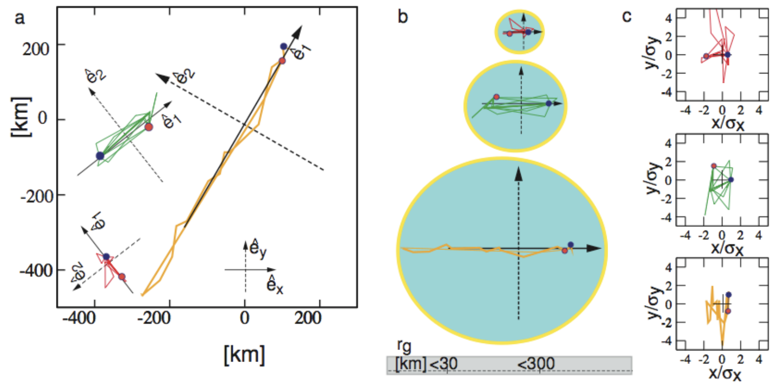


Figure 32: Example of how to transform the user trajectories in a common reference frame. *a)* Initial trajectories of three users and their principal axes($\hat{e}_1, \hat{e}_2$). *b)* Each trajectory is rotated to align $\hat{e}_1$ with $\hat{e}_x$. *c)* Positions (x, y) are scaled as $(x/\sigma_x, y/\sigma_y)$ after which the different trajectories have a quite similar shape. (From [GHB])

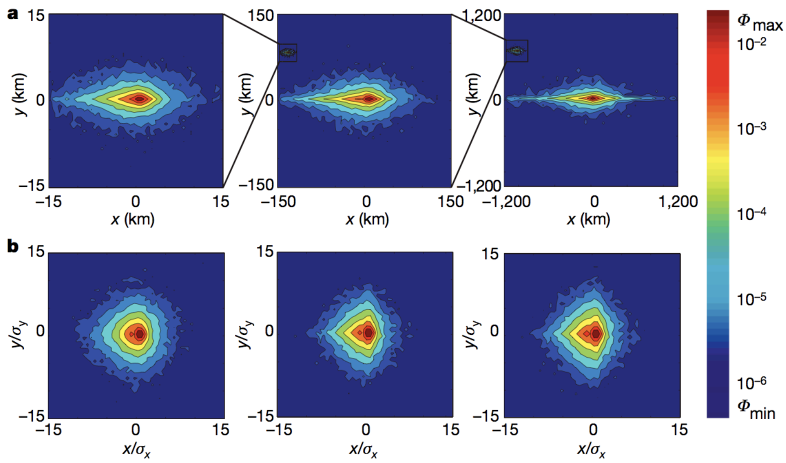


Figure 33: The shape of human trajectories. *a)* The probability density function $\Phi(x, y)$ of finding a mobile phone user in a location (x, y) in the user's intrinsic reference frame. The three plots, from left to right, were generated for 10,000 users with: $r_g \leqslant 3$, $r_g \leqslant 20$, $r_g \leqslant 30$ and $r_g \leqslant 100$ km. The trajectories become more anisotropic as r_g increases. *b)* After scaling each position with σ_x and σ_y, the resulting has approximately the same shape for each group. (From [GHB08])

3.4 CONCLUSION

In this chapter we surveyed the various approaches that can be related to mobility modeling for wireless network context, including microscopic mobility models and other relevant trace analyses. The models are then divided according to their founding principles in two categories - synthetic (Section 3.1) and empirical (Section 3.2).

Synthetic models aim at model simplicity and mathematical tractability and can be further subdivided to analytical and physically-based. The synthetic analytical models are based purely on mathematical and/or æsthetical considerations, whereas the physically-based ones aim at modeling some established characteristics.

Empirical models are based on heavy collections of real-world data, and depend on their type and granularity. Depending on the type of data they can be divided into map-based and trace-based. The latter can be further subdivided into coarse-, fine- and encounter- trace data depending on the temporal and spatial granularity of the data and on its type.

We have surveyed several of the existing mobility models, which we have categorized in a cognitive map (Fig. 3) depending on the number of parameters, which is proportional to their complexity and scenario specialization. The models were selected based on their popularity, innovation, and scope of application and founding principles as to present the full range of existing mobility modeling practices and approaches. For the sake of brevity, some of the models are presented in Sections A.2 and A.3 of the appendix.

Finally, we have surveyed several types of data traces which were (or could be) used for mobility modeling. We have put an emphasis on these traces because they are an important source of data for validating and designing new mobility models. We presented WLAN association, Bluetooth encounter, and GSM traces, which indicate some important characteristics of human movement patterns in several scales (country, city and university campus), both concerning relative (encounter) and absolute positioning.

In table 3.4 we have summarized the mobility models we surveyed in this chapter and the appendix. The ease of use is based on the number of parameters and the difficulties of generating them - it defines how much work should be performed for the model to be usable. The parameter accessibility shows if the role of the parameters is straightforward and if it is clear in what way they will affect all aspects of node movements. The "Analyzed" column indicates if some of the mathematical properties of the model were studied. However, some other important characteristics may still need to be explored. The "Group mobility" shows if the model is capable of generating group movement patterns. The usage of movement constraints is provided in the "Constrained Movement" column. The column "Data based" indicates if the model was created based on characteristics extracted from empirical data, while the "Validated" field shows if the model was validated against a real-world context. The distribution of the Aggregated ICT is indicated in the next column, and can be power law, exponential and power law with exponential tail. The possibility to create movement patterns whose base characteristics vary among the nodes is indicated in the "Heterogeneous movement" columns. The "Flexible" column denotes if the model can represent significantly different types of movements by tweaking its parameters. This column is not formally defined and is based on the number of different models that can be represented as the model in question.

An ideal mobility model should possess all these characteristics, be easy to use and provide power law with exponential cutoff ICT (or other, if the context is clearly defined). Unfortunately, there is no such model, with only several physically-based models approaching the goal. They are however limited to certain types of scenarios or are more difficult to configure. In the next chapter (Chapter 4) we are going to address the issue of creating models which possess all these properties with minimal complexity. The architecture we propose divides a model in several layers, and each of the layers may be simple and satisfying all characteristics met by the best models from this table, and in the same time provide aspects which are missing, such as flexibility, based solely on the features of the architecture.

Table 1: Summary of the existing mobility models.

Model	Ease of use	Accessible parameters	Analyzed	Group mobility	Constrained movement	Data based	Validated	ICT	Heterogeneous movements	Flexible
Analytical synthetic models										
RWP	e	x	x				x	exp		
RW	e	x	x					exp		
RD	e	x	x					?		
Smooth RD	m	x	x					?		
Location Dependent	m	x						?		x
Smooth RD	e	x	x					?		
Obstacle	m	x			x			?		x
Delaunay	m	x			x			?		x
City section	e	x			x			?		
Manhattan	e	x			x			?		
Freeway	e	x			x			?		
Virtual track	e			x	x			?		
RPGM	m	x		x				?		
Structured Group	m			x				?		x
Exponentially correlated	e		x	x				?		
Physically-based synthetic models										
Ant colony	e	x						?		
Pragma	e	x	x					?	x	
Clustered	e	x	x					exp	x	
SIMPS	m	x	x					pexp	x	
Community-based	e	x	x					pow	x	
TLW	e	x	x			x	x	pexp		
HRW	e	x	x			x	x	?		
SLAW	e	x	x			x		pexp	x	
Empirical models										
UPF	h	x	x		x		x	?		
WLAN Kim	h					x	x	?	x	x
WLAN Tuduce	m	x				x	x	?		x
WLAN Statistical	h	x			x	x	x	?	x	x
Activity-based	h	x				x		?	x	
WWP	m	x			x	x		?	x	x

The ease of use has three levels - *e*asy, *m*edium and *h*ard. The ICT also has three levels - *pexp* (power law with exponential cut-off), *pow* (power law) and *exp* (exponential). All other columns indicate with *x* if the value is yes, and with empty field if it is no.

Part II

MOBILITY MODEL FRAMEWORK

4 LAYERED ARCHITECTURE

This chapter introduces our proposition - the generic layered mobility model architecture LEMMA. The architecture is aimed at addressing the issues pointed in the introduction (Section 1). More concretely, as we show in the analysis section of this chapter (Section 4.4), our architecture can be used as basis for creation of mobility models which are:

- Accessible and easy to use.
- Sufficiently detailed and easy to understand.
- With accessible parameters.
- With studied mathematical properties.
- Verified and validated.

The fundamental principle of the architecture states that a model is divided into several *layers* (fig. 35), each layer having distinct, specific 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 has 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 group behavior may be simulated by sharing layer implementations across a set of nodes. 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 or prove specific mathematical properties of the whole model based only on properties of its constituent layers.

Section 4.1 presents the basic relationships between environment and movement processes. The environment components of the architecture are presented in Section 4.2, while the node movement process separation is detailed in Section 4.3. A more detailed discussion of the implications of the fundamental principles can be found in Section 4.4. The mathematical properties of the architecture are studied in Chapter 5.

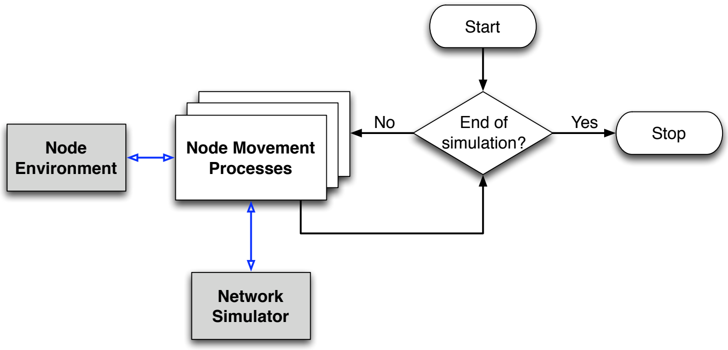


Figure 34: Schematic illustration of the interaction between movement process and node environment.

4.1 NODE ENVIRONMENT AND MOVEMENT PROCESSES

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 environment may be considered as parameters common to all movement processes. The movement processes specify node's movements as a function of their environment and simulation parameters (illustrated in fig. 34).

In the majority of cases, there is only one movement process, which governs the movement of all nodes in a simulation (e.g. RWP). However, one may find scenarios where each node has its own movement process (as in multi-agent simulations), processes that govern the movement of all nodes within a given subset of the simulation area, etc.

4.2 ENVIRONMENT

The simulation environment is one of the first elements that has to be fixed in any scenario. Having a set of unified environment components not only facilitates the definition of such scenarios, but also makes possible the definition of abstract layer interfaces. By studying the various existing mobility models we found that there are four basic types of such components, which we have named *simulation area*, *zones*, *constraints* and *movement influencing factors*. The rest of this section details each of these parts.

4.2.1 *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 $(p_0, \ldots, p_N)$. The action to be taken if a node tries to move outside the simulation area depends on its *border behavior*. In [Bet01a] Bettstetter summarized the bounce-back, wrap-around and delete-and-replace behaviors. When a node attempts to move outside the simulation area, the bounce-back effect changes its trajectory in such way as if the node is reflected from the border, the wrap-around moves the node to the opposite border, while the delete-and-replace moves it somewhere on the simulation area. The wrap-around border behavior makes a 2D rectangular simulation area equivalent to a torus.

4.2.2 *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. Furthermore, the need of a high-level addressing is obvious in some of the models, where the nodes move between some large domains differently than within the domains (as in [BGB01] for example). To implement this kind of abstraction, the simulation area can be divided into multiple zones. A *zone* is a set of points with a (possibly empty) set of attributes. For example, if in a given scenario the simulation area reflects a real-world city, one may use rectangular zones to represent the buildings, each having an attribute defining its purpose (if it is a restaurant, a library, ...). The way zones are defined allows the straightforward specification of zone relations, such as *intersection*, *union*, etc.

4.2.3 *Constraints*

Node movement is directly affected by the movement *constraints*. They depend on the movement process itself, e.g. a constraint may be in the form of a graph, a set of rectangles. In order to achieve meaningful mobility patterns it is reasonable to expect that in a given scenario the constraints and the zones should not be placed regardless of each other (e.g. if a zone has to be visited by a node there must exist a way for it to reach the zone).

4.2.4 *Movement influencing factors*

Finally, the low-level aspects of the movement depend on the *movement influencing factors*. They are also specific to 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.

4.3 MOVEMENT SUBPROCESSES

Node movement processes govern the movement of the nodes in the simulation environment. In *LEMMA*, a general movement process is divided into five layers, called *strategy*, *mapper*, *tactic*, *dynamics*, and *stay* (as shown in fig. 35). These layers communicate via simple, strictly defined interfaces in a top-to-bottom manner (e.g. the interlayer communication is unidirectional). The purpose of the different layers is given hereafter along with the definition of the layer interfaces and some simple usage illustrations. A detailed survey of the layers found in the literature and more in-depth examples of the architecture use-cases are given in Section A.1.

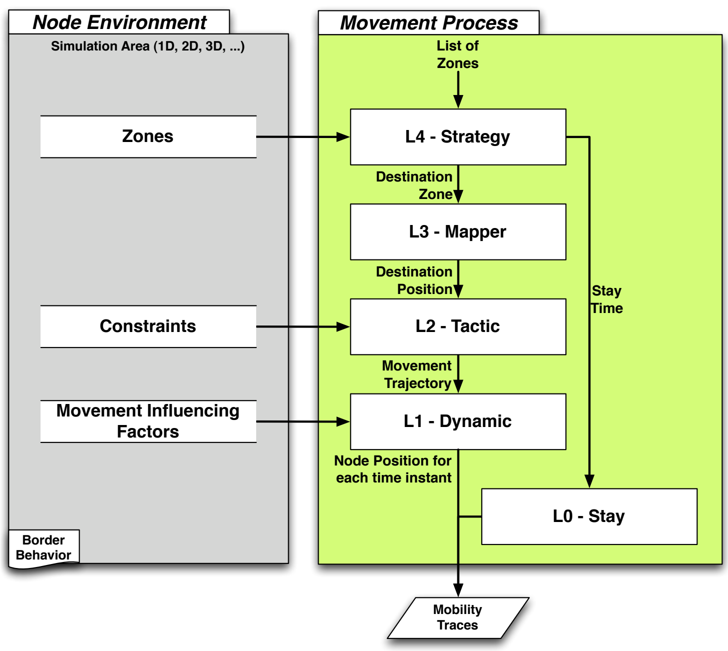


Figure 35: Node environment entities and movement process layers.

4.3.1 *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. As an example, a simple strategy may randomly choose the next zone and stay time.

4.3.2 *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. For example, a basic mapper may just select a random point from the zone, ignoring its shape or attributes.

4.3.3 *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. The simplest tactic is a linear movement from the departure to the destination point.

4.3.4 *Dynamic*

Finally, the movement *dynamics* specifies the speed and the acceleration, and possibly some small deviations from the trajectory that has been defined by the *tactic*. This layer corresponds to human behavior, e.g. people change their speed depending on their environment and do not follow blindly a preset average speed. Furthermore, it allows for complex mobility behavior to be simulated, such as adding traffic lights to the simulation, without touching the upper layers. An example dynamic may stipulate constant movement with fixed speed and no trajectory deviations.

4.3.5 *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, its movement is governed by the movement process defined in the *stay* layer. 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 [BGB01]), etc. The process set in this layer can be described with the same layered architecture.

4.4 ARCHITECTURE ANALYSIS

The two fundamental principles around which is built LEMMA are the functional differentiation of the layers and the simplified, fixed interfaces used for layer communication. This section provides examples of some of the implications of these principles, including advanced model creation and layer validation. The mathematical properties of the layers which could be studied with the help to the same principles are given in Chapter 5.

The architecture is founded on the principle that a model can be represented with several independent layers which communicate via simple interfaces. This was made possible by standardizing the environmental components and fixing the layer scopes. A direct consequence of these design decisions is the facilitation of some commonly performed actions such as model verification and validation. Most importantly, all implications of *LEMMA* are result of the generic layer definition and do not require layer specialization, i.e. the approaches described hereafter can be used directly by any existing layer.

Here, we are going to analyze some of the direct ramifications of *LEMMA*'s founding principles. Although some of them may seem to originate from a subset of architecture's functions, the full potential and the complete set of benefits is a result of the interplay of all characteristics of *LEMMA*. In another terms, the architecture as described earlier in this section is the minimal framework providing the totality of possibilities discussed hereafter.

4.4.1 *Scope*

Each of the layers targets a specific level of node movement. For example strategies deal with high-level movement behavior, while dynamics determine the low-level, instantaneous movement patterns. It is there-

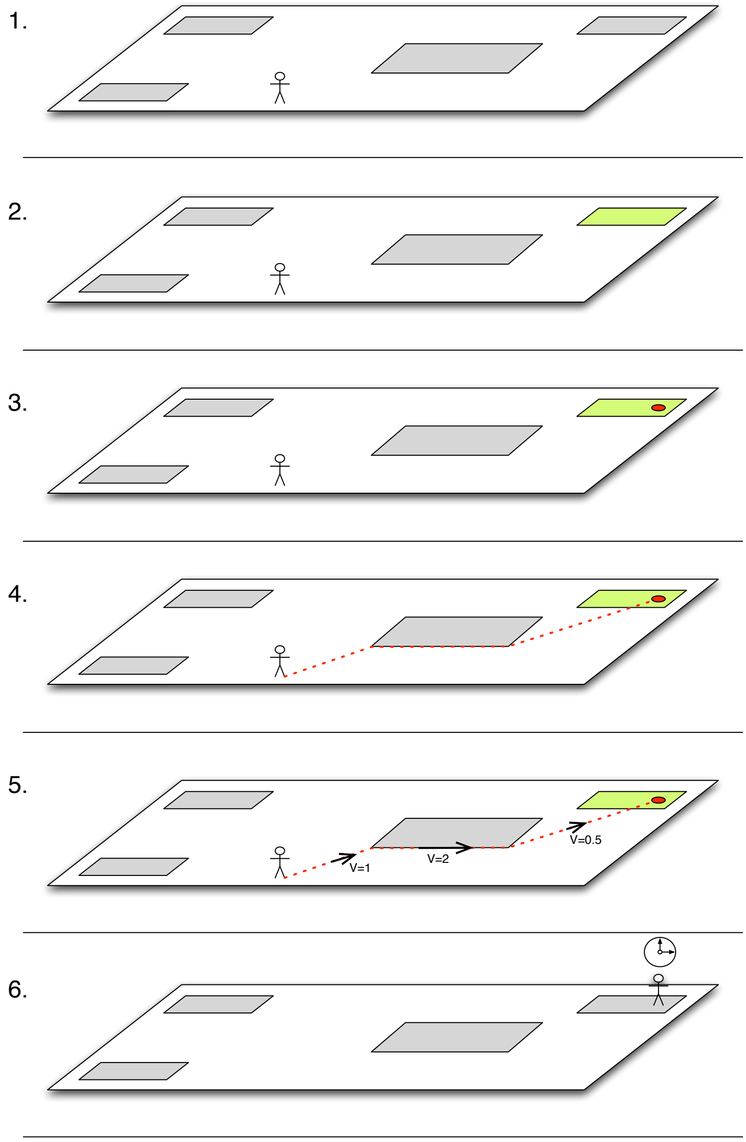


Figure 36: Mobility trace generation example. The strategy selects the destination zone in **2**, then the mapper chooses the exact position in **3**, followed by the tactics, which selects the route to be followed in **4**. In **5**, the dynamics determines the speed at each point, and finally in **6**, the stay layer specifies what should be done in this zone (in this case – pause). 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).

fore easier to understand which parameters modify the coarse-grained node behavior, and which deal with fine-grained movement details. Furthermore, the scope differentiation incites better model understanding during the model conception and facilitates its comprehension. Additionally, presenting a model in such manner assures that all necessary elements of it are fully described, and in the same time simplifies the introduction of new principles.

4.4.2 *Layer Generalization*

In addition to the minimal layer interfaces, abstract environment description enables generic layer definition, i.e. layers are defined only with their essential environment characteristics while unnecessary dependencies are avoided. For example, if a strategy does not use any location-dependent information it can be applied to 1D, 2D or 3D simulation area without any modifications. This kind of flexibility is important as we have witnessed the same type of movement principles being redefined in several propositions only because of differences in the underlaying scenarios.

4.4.3 *Verification*

When a model is divided into several independent modules, the implementation of each module can be verified separately from the others. In the cases when some of the layers are reused, only the newly developed have to be verified. Additionally, because of the standardization of the layer interfaces, some common, implementation-independent verification routines can be realized. For example, defining generic unit test [otICS99] which trial key layer characteristics makes them applicable to all layer implementations, e.g. it is possible to automatically verify some of the aspects of the layers. Furthermore, because the implementation of each layer is independent of the rest, it is possible to test rare events which would otherwise be hard to verify.

As a simple example, a unit test verifying the compliance with some of the basic traits exhibited by every *tactic* layer could check if all generated trajectories contain the starting and ending points and if they are entirely contained in the simulation area.

4.4.4 *Validation*

There exist an abundance of studies on various mobility-related phenomena, which map directly to the different layers of our architecture. They may be considered as an evidence of the universal character of the layers. Most importantly, the conclusions and the empiric evidences of these studies can be used to define general validation criteria, relevant to the specific layers. For example, based on the standardized layer interfaces, one could define validation contexts for each of the layers in a way resembling the realistic contexts introduced earlier, thus providing a general way of validating the behavior of a model on several stages.

Here, we are going to briefly describe some of the studies which could be used for layer validation.

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 $\mathtt{cell} \leftrightarrow zone$, the majority of them can be considered as direct implementations of the strategy layer. For example, Bhattacharya and Das [BD99] 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. [SKJH04, Son08] and compared to Markov-based predictors. Another class of relevant work include the OD studies performed by the traffic engineering researchers, e.g. [Abr98]. 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 [AG92].

The mapper layer, on the other hand, has not received as much attention, probably because it acts as a binding layer between the strategy and the tactic, and thus cannot exist on its own. An example of model using a mapper is the one presented by Minder et al. [MMLR05]. The mapper is based on office's layout and determines the exact location as a function of the room type, such as sitting on one's desk, or grouping in the middle of a conference room. In this case, one may consider the vast amount of work done by architects and designers as studies on the way the mapper should be configured. Furthermore, there exist synthetic methods to generate realistic office layouts, such as the genetic algorithm approach described in [NHH$^+$06].

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 [Gol95] (i.e. taking path a when going from point A to point B, and a completely different path b in the other direction). There are evidences that people use cognitive maps instead of remembering the exact streets layout [Gol99], which also fits nicely to the separation of tasks between the tactic and the dynamic layer (see [MTF04] for construction of such map).

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 [JWZ01], the Intelligent Driver Model [THH00], the bounded rational driver model [LWM03], the Human Driver Model [TKH06], etc. Pedestrian movement requires other types of models, e.g. we can find multiple propositions addressing emergency evacuations ([Kul05]), flow modeling [Hoo04] (e.g. the intersection of two streets, the merging/splitting of two corridors, etc.), repulsive and attractive properties of different objects (such as walls, shops, ...), and many more.

4.4.5 *Mathematical Tractability*

The approach chosen for the definition of the environment components, combined with the simple layer interfaces provide the possibility to analyze the properties of the mobility models defined with LEMMA by only knowing the properties of the separate layers (as shown in Chapter 5).

4.4.6 *Universal Model Representation*

In spite of the small number of restrictions on the layers, and the simplicity of the interfaces over which these layers communicate, LEMMA is able to represent any mobility model in a generic, non-trivial way (as we proved in 5.4.1). Additionally, we have synthesized the various environment components (Section A.1.1), strategies (Section A.1.2), mappers (Section A.1.3) and dynamics (Section A.1.5) which can be used to construct the majority of existing synthetic and empirical models (surveyed in Chapter 3 and Sections A.2 and A.3).

These analyses and model representations testify to the universal characteristic of LEMMA and its potential to be used as fundamental framework for mobility model definition and analysis.

4.4.7 *Layer aggregation*

Fixed layer interfaces provide the possibility to obtain new layers by aggregating already existing ones of the same type. This can be achieved with the help of a limited set of predefined *adaptors*. If the possibility to recombine existing layers into new models provides *vertical* modularity, layer aggregation can enhance the architecture by allowing both *vertical* and *horizontal* layer modularity. Most importantly this feature is a direct consequence of the framework itself and does not add any constraints or requirements on the layers themselves, e.g. any *LEMMA* layer can be aggregated regardless of its nature. Finally, this functionality does not increase the complexity of the architecture, but increases its expressiveness. The aggregated layers can be submitted to existing verification and validation procedures.

We are going to provide layer aggregation examples in order to illustrate the *vertical* and *horizontal* layer modularity. Vertical modularity is demonstrated with strategy layers and horizontal modularity is shown with dynamic layers.

Defining relations between zones such as "contains" or "intersects" is straightforward as they are sets of points. With the help of these relationships, the output of a given strategy layer can be processed by some *strategy adaptors* and be provided as input to another strategy layer, e.g. *vertical* modularity. This type of aggregation is schematically illustrated in Fig. 37 where two adaptors are used one for the zone and one for the time parameter.

The zone adaptor selects a list of zones as a function of the destination zone provided by the upper strategy. The list is then passed on to the lower strategy as a parameter. An example zone adaptor may select all zones that are fully contained in the destination zone. The time adaptor aggregates the multiple stay times into a single stay time output. There may be many types of time adaptors, such as always taking one of the stay times (e.g. the last one), taking the average of all times, etc.

Such adaptors may be defined for all types of layers because of the standard layer interfaces. An aggregated dynamic layer can be used as an example for *horizontal* layer modularity. Dynamic layer output can be represented as velocity vector for the node for the given time instant, thus any of the standard vector operations can be then used to aggregate the output of two or more such layers. In Fig. 38 is sketched the scheme of layer aggregation with *dynamic adaptor*. For example, if the aggregator outputs the average of the provided vectors it could

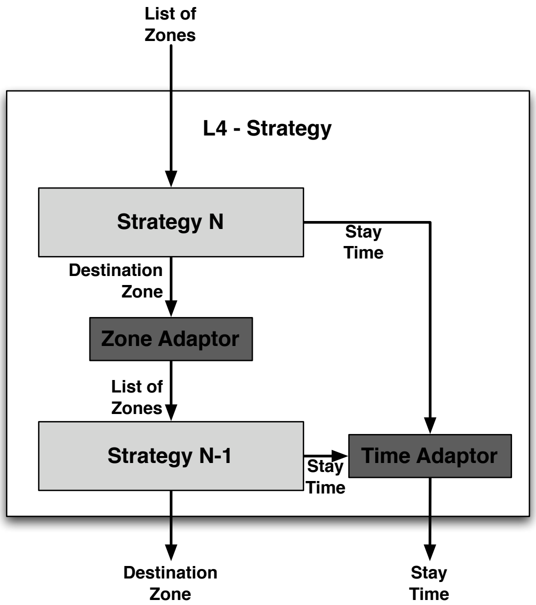


Figure 37: Vertical layer aggregation. Combining several strategies into a single one is possible with the help of zone and time adaptors.

be used to simulate minor deviations from the main trajectory, e.g. a person crossing the street to look at a shop or to avoid a crowd, while still moving towards his/her destination.

4.4.8 *Heterogenous models*

A node may be required to change its movement patterns during the course of the simulation, e.g. to simulate pedestrians getting on and off busses, taxies or other forms of transportation. 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 criteria, like current time, zone, surrounding nodes, etc. (demonstrated schematically in fig. 80). The possibilities of this type of "simple" model exchange are augmented with the possibility to substitute individual layers. Changing only the dynamic layer for example could alter the low-level behavior details, without modifying the high-level movement process (Fig. 40).

4.4.9 *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. 41. This facilitates the creation of a group-like behavior, e.g. we may define the RPGM 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).

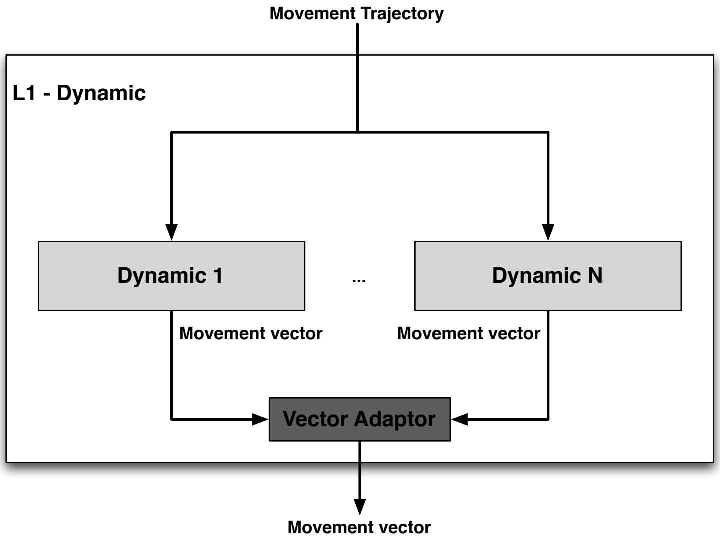


Figure 38: Horizontal layer aggregation. Combining several dynamics into a single one.

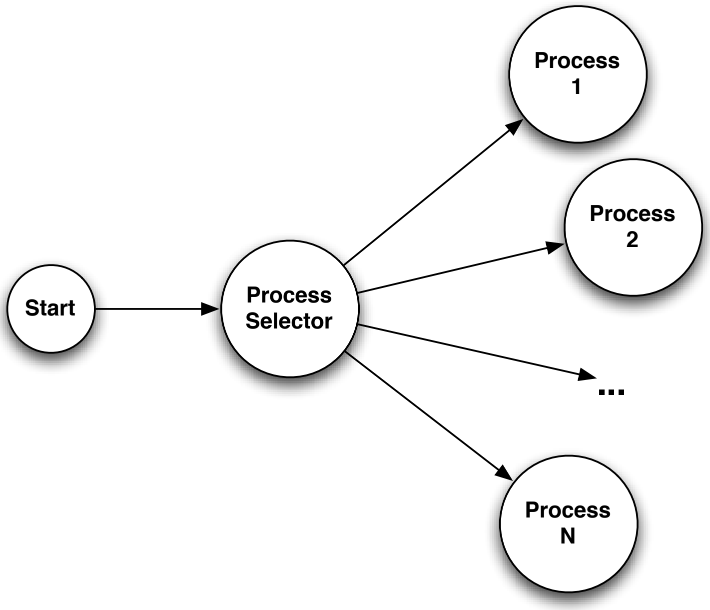


Figure 39: Multiple processes may be dynamically switched to create a hybrid node movement process.

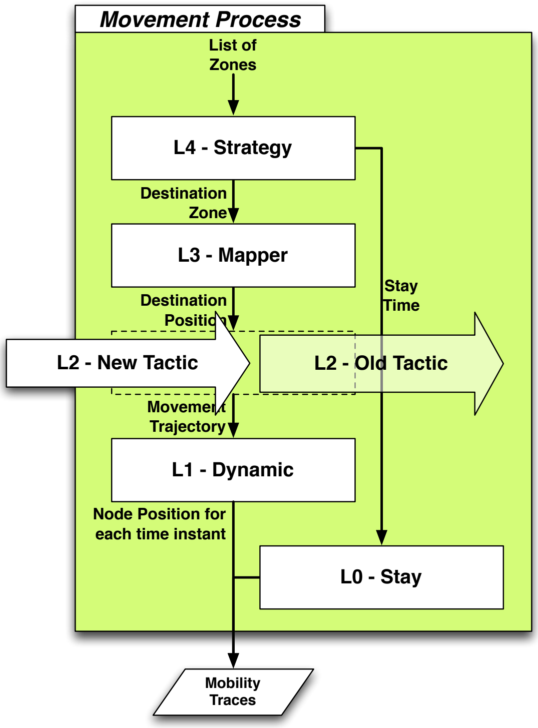


Figure 40: Fine-grained hybrid node behavior may be achieved by changing a single layer of its process.

4.4.10 *Practical Framework*

A very important characteristic of each mobility model is its availability and ease of use. We speculate that this is one of the important factors which make the RWP the most used mobility model [KCC05].

Important characteristic of our architecture is that apart from being an universal basis for theoretical works and model definitions, can also be a practical framework which can be used for the simulation of the specified models. In order to support this, we have developed a fully functional simulator. We have taken a pragmatic approach towards its realization - e.g. we have followed its principles, while in the same time simplifying some of the assumptions, without sacrificing generalization. As an example, we have assumed that the simulation area is always a 3D space, and regard the 2D as a special case. Furthermore, all object have been created as implementations of generic interfaces, so in case any of these assumptions is challenged in the future, new realizations can be added without altering the core components of the architecture.

The simulator is implemented by using SimPy [sim], an open-source, object-oriented, process-based discrete-event simulation language based on standard Python. The usage of Python as a base language has many positive implications on its flexibility and ease of use. Here is an example illustrating the simplicity of creating a layered mobility model once the different layers have been defined:

```
mp = LayeredMovementProcess(node, environment)
mp.strategy = UniformStrategy(min_pause, max_pause)
mp.mapper = random_mapper
mp.tactic = linear_tactics
mp.dynamic = ConstantMovementDynamic(min_speed, max_speed)
```

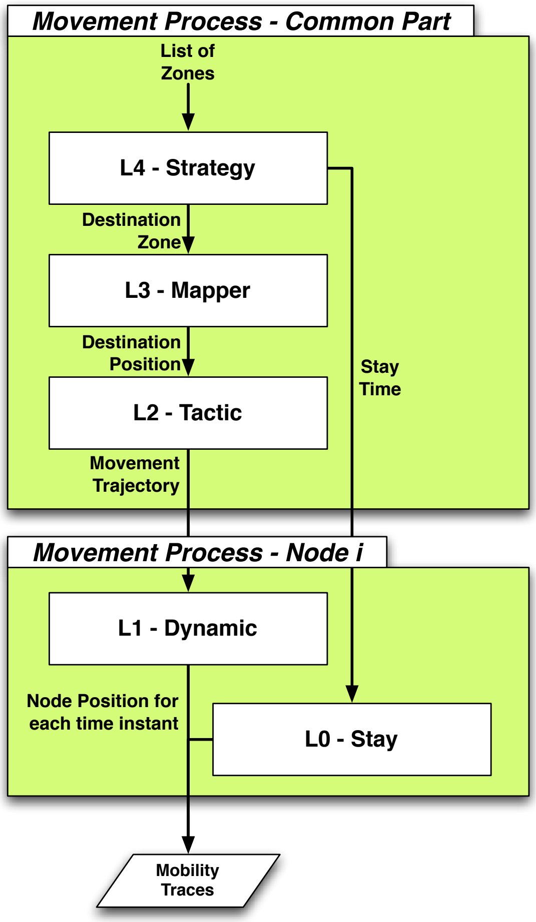


Figure 41: Several nodes may share the same layer instances. Here the nodes have common strategy, mapper and tactic layers. All nodes receive the same input from the strategy and tactic layers.

```
mp.stay = PauseStay()
```

The implementation complexity of the different layers may vary, but once a layer is developed, its usage becomes trivial. This kind of separation provides mobility model users with the flexibility to use any combination of already existing layers regardless of their author or base principles. We have detailed the implementation of the simulator along with its architecture, comprising components and usage examples in Section B.

4.5 RELATION TO OTHER FRAMEWORKS

Other mobility model frameworks targeting the same goals have been proposed. Here, we are going to briefly describe two of the most popular ones. We provide a correspondence between the various components of these frameworks and LEMMA, which indicates how a model described in any of these frameworks can be represented in LEMMA.

4.5.1 *UOMM*

In a series of publications [Ste02, SHB+03, SMR05] Stepanov et al. have presented and applied the *User-Oriented Mobility Model* (UOMM). It separates the movement generation into three distinct models - spatial, user trip and movement dynamics. The environment contains points, lines and polygons, which represent roads and points of interest. The trip generation specifies the movement trajectory, while the movement dynamics specify how momental speed is calculated. This design has several points in common with *LEMMA*, so we tried to use the same terms, wherever possible. However, UOMM does not define separation between the high-level decision making process and the low-level trajectory generation process. Additionally, we have identified the different sub-entities of the environment, the possible ways of strategy aggregation and the creation of hybrid solutions and group mobility models, which are crucial for the wider application of the architecture. The relations between the two architectures may be seen in fig. 42.

4.5.2 *ORBIT*

The ORBIT Mobility Framework, defined by Ghosh et al. in [JGQ04, GPQ05, GBNQ06], is based on the observation that a person tends to visit different sets of places, each set being called an *orbit*. The specific set of places to be visited (i.e. the selected orbit) may change depending on various factors, e.g. "home → work → cinema → home" for a workday. The places that can be visited are called *hubs*, and are assumed to be rectangular. The framework specifies a hierarchical stacking of orbits, where each level is regarded as a black box for the other levels. The lowest level confines the movement of a node within a single place, where it moves following a predefined model. All higher levels take a set of places as input, and output a set of one or more places - the level directly above the lowest one should output a single place, and the way to move to it. Thus, the process of generating the movement may be described as *orbit selection → selection of a sub-orbit → . . . → selection of a sub-orbit → selection of a single hub → simple movement in the hub*. Compared to *LEMMA*, the given design does not specify a

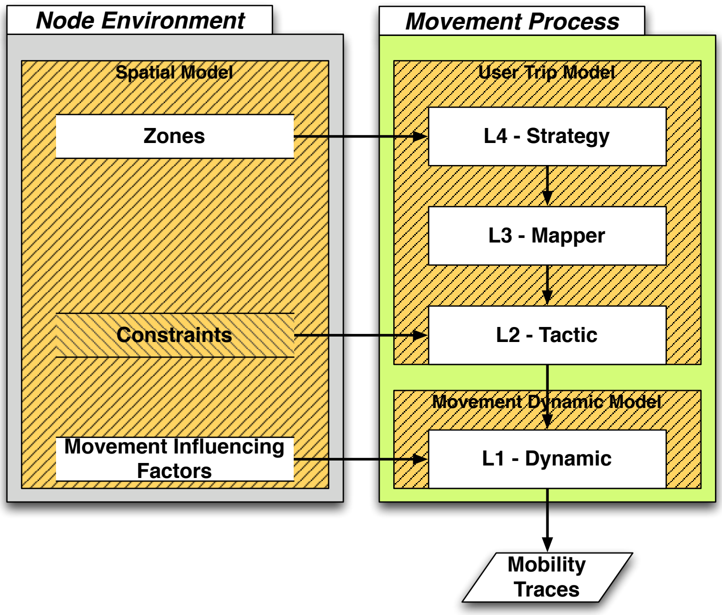


Figure 42: Relation between the User-Oriented Mobility Model and *LEMMA*. Parts presented in UOMM are in striped orange.

detailed environment specification. All layers found in our approach are identified with a greater separation of the functions, notably having distinct mapper, tactics and dynamics layers. Most importantly, the principles behind ORBIT are aimed at separating the movement in two types of processes - one handling the global movement, and the other - the local one, whereas our separation provides a gradual high- to low- movement refinement, which breaks both the global and the local movements into functionally distinct layers. The direct mapping from ORBIT to our proposition is shown in fig. 43.

4.6 CONCLUSION

In this chapter we introduced our proposition - the Layered Mobility Model Architecture (LEMMA). The architecture follows few simple core principles that provide an abundance of possibilities, which have been analyzed.

One of the main principles specifies the standardization of node environment. It contains the space where the nodes exist (*simulation area*) which contains named subsets of points (*zones*). Additionally, it hold the factors influencing the movement behavior (*constraints* and *movement influencing factors*).

The second core principle states that the node movement process is divided into five distinct layers communicating via simple interfaces in unidirectional, top-to-bottom manner. The layers are with strictly distinguished scopes, each responsible for different stage of the movement generation. The *strategy* layer is the high-level decision-making process, which defines the movement of the node on the *zones* level, e.g. coarse-grained movement. The *tactic* is the route selecting layer, which defines the exact trajectory to be followed by the node for reaching a destination point from node's current location. The *mapper* acts as a

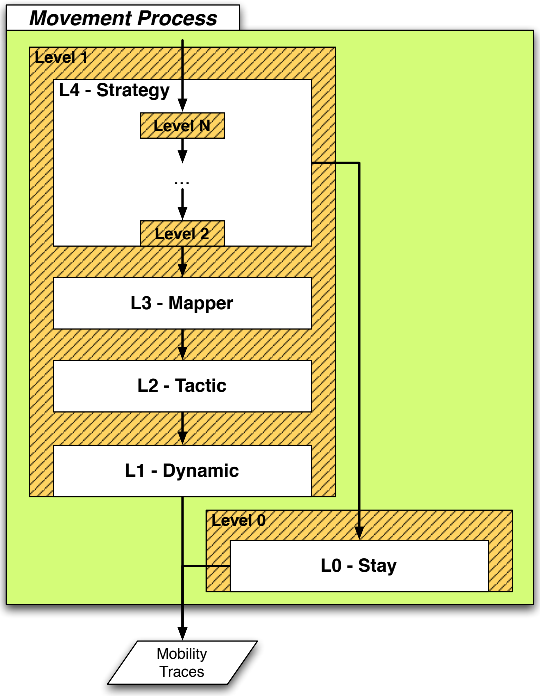


Figure 43: Relation between the ORBIT Framework and the Layered Mobility Model Architecture. Parts presented in ORBIT are in striped orange.

translation layer between the coarse-grained movement *strategy* layer and the path selecting *tactic* layer. Once the route to be followed is determined, the *dynamics* layer adds the temporal dimension of the movement, i.e. it specifies the speed at each point of the trajectory. Finally, after reaching the destination point, the *stay* layer takes over and governs the movement of the node for a duration determined beforehand by the *strategy*. The division of node movement process into layers, approach used in many other existing systems such as the OSI/ISO model, is made possible by the standardization of the environment.

Layer definitions and inter-layer interfaces were defined in a generic way, which incites newly defined layers to be also general, provided in manner as broad as possible. This is in contrast to the existing approaches, which often focus on the scenarios at hand and miss the opportunity to provide wider scope of application.

More importantly, the standardized interfaces provide the possibility to define implementation-independent verification and validation procedures. Indeed, once such procedures are specified for a given layer, all existing and future layer realizations can be submitted to them without modification or adaptation of the layers, or the validation/verification methods.

Several advanced techniques were also presented, which allow the definition of heterogeneous or group models, and the aggregation of existing layers into new ones. Heterogeneous models can be comprised of several simple movement processes which are exchanged, or a single model with individual layer replacement. Additionally, sharing several layers across a set of nodes may be used to generate group movement patterns. New layers can also be obtained by combining several existing ones with the help of layer adapters. Most importantly, these approaches come at no additional cost to model creators and users, as the flexibility lies entirely in the framework structure, and does not impose restrictions on the individual layers.

In order to provide a tangible evidence of the feasibility of LEMMA we have developed an open-source mobility simulator and have provided a wide set of layers defined in various synthetic and empirical models found in the literature. They have been outlined in this chapter and are given in detail in the appendix.

We have given a comparison with other existing general mobility model frameworks, and have provided the correspondences of their components. The interplay of the core principles of LEMMA provide greater set of possibilities for the creation, verification and validation of simple and complex (e.g. heterogeneous) models. Creating group mobility patterns is a unique feature of LEMMA among the other generic frameworks. The combination of these aspects provides a platform for the definition of models which are easy to describe, use, modify, verify and validate - characteristics having utmost importance for the usability and acceptance of a mobility model.

These points testify to the unique character of LEMMA, its simplicity, flexibility, and feasibility. In this chapter, we analyzed them in detail, and provided evidence that models based on LEMMA have many of the most important mobility model characteristics we discussed in the introduction (Section 1). We continue this study in the next chapter (Chapter 5), where we provide the mathematical foundations of these LEMMA-based models. Most notably we provide formal proofs of the universal character of LEMMA and theorems showing that it is possible to obtain global model knowledge by only having local layer characteristics.

5 MATHEMATICAL FOUNDATIONS

The architecture introduced in the previous chapter (Chapter 4) can be used for the definition of empiric and synthetic models. It is even possible to imagine scenarios where some of the layers are purely synthetic, while others are data-driven, thus creating the possibility to define mixed types of mobility models.

An interesting question regarding the framework is wether if it is possible to deduce some of the mathematical properties of the constructed mobility models by only knowing the properties of the constituent layers. In this chapter we answer positively to this question. We start by giving a formal definition of a mobility model, and move on to give sufficient conditions for the final model to exhibit essential mobility model characteristics such as stationarity.

5.1 MOBILITY MODEL

We start this section by giving a formal definition of a mobility model. Afterwards, a formal proof will be given, showing that any mobility model can be represented with LEMMA.

Definition 5.1.1. Informally, we can say that a *mobility model* is a mathematical or computer model which can be used to analyze or simulate some (or all) of the aspects of entities' *movements*.

Definition 5.1.2. If the model is deterministic function, 5.1.1 can be formalized as:

$$f_m : \Omega \times T \to E \tag{5.1}$$

where T is a an index set which may be considered as the time (discrete or continuous), E is the destination space, and Ω is the parameter space of the model. Additionally, the spaces $(\Omega, \mathcal{F})$, $(E, \mathcal{E})$, and $(T, \mathcal{T})$ should be measurable, and f_m should be $\mathcal{F} \otimes \mathcal{T}/\mathcal{E}$-measurable, where $\mathcal{F} \otimes \mathcal{T}$ designates the product σ-algebra over $\Omega \times T$.

Definition 5.1.3. If the model is stochastic, 5.1.1 is formalized as:

$$\mathbf{X}_m = \{X_t | X_t : \Omega \to E, t \in T\}. \tag{5.2}$$

with T, E, and Ω having the same meaning as in 5.1.2. However, $(\Omega, F, \mathbb{P})$ should be probability space, and X_t should be F/T-measurable for every $t \in T$.

The above definitions are very natural, simply stating that a mobility model, given some parameters, produces a trajectory in the required space $T \to E$ (i.e. associates a point $e \in E$ for each time instant $t \in T$). Additionally, we require that all functions or stochastic processes are well-behaved, i.e. are not pathological cases. Throughout this section, we are going to be stating explicitly what that means, but it is sufficient to take into consideration that we are only using natural constructs, which could be omitted if we weren't concerned with the exact proofs of the theorems. As an example, every time a function is required to be measurable we explicitly give the concerned σ-algebras. Moreover, we have defined the model in an abstract way, but in most of the cases $E \equiv \mathbb{R}^2$ and $T \equiv \mathbb{N}$.

Saying that a space is measurable means that we have defined a σ-algebra on it, that is - we have declared which subsets are "good", and whose "volume" we can calculate. For example, when considering $(\mathbb{R}, \mathfrak{B})$ (the real line $\mathbb{R}$ with the Borel σ-algebra $\mathfrak{B}$) we can use the Borel measure to calculate the length of the interval $[0; 1]$ (which belongs to the "good" sets we can measure) in the natural way, i.e. $1 - 0 = 1$.

Note, that even if this definition seems to capture only single node movements, this is not necessarily the case. Consider the movement of a group of N nodes, where the location of each member depends on the position of the others. Lets define the function $f_d : T \to D$, where D is an $N \times N$ matrix whose main diagonal is $\bar{0}$. The space of all such functions F_d exists and is non-empy. It is possible to define a mobility model which generates the traces of a member of the group, taking in account the distances to all other members of this group, which are given for each moment $t \in T$ by the distance matrix $f_d(t)$. The distance can be measured with the standard Euclidean distance, in which case all distance matrices would be symmetric. Inter-node communication can also be modeled with the help of functions indicating message arrival. Of course, implementing a simulation by blindly applying these principles would be very difficult, or even impossible (e.g. performing simulations for all possible distance matrices and message exchanges), but the approach allows us to analyze a broad spectrum of models with simple tools.

5.2 DETERMINISTIC MODELS

A model can be defined by following the LEMMA Model Representation (LMR). This means, that it has been divided into layers, each layer having the functions defined in Section 4. Here, we are going to give a formal definition of what is meant by LMR when the mobility model is a function, or a stochastic process.

First, we start by defining the set of all zones taking part of a scenario $\mathcal{Z} = \{Z_i | i = 0 \dots k; k < \infty, Z_i \subseteq E, Z_i \neq \phi, Z_i\text{- closed}\}$. This set depends entirely on the scenario creator, and should be non-empty. Selecting the zones to be closed sets is a convenience which allows us to select points from their borders. Also, we will denote the union of all zones as $\mathfrak{Z} = \cup\mathcal{Z}$, which means that $\forall e, i : e \in Z_i \Rightarrow e \in \mathfrak{Z}$ and $\forall e \exists i : e \in \mathfrak{Z} \Rightarrow e \in Z_i$.

Then, let us define for convenience $\mathcal{G}$ as the set of all $\mathfrak{B}([0; 1])/\mathcal{E}$-measurable functions $\mathcal{G} := \{g | g : [0; 1] \to E\}$, and $\mathcal{D}$ as the set of all strictly monotone increasing $\mathfrak{B}([0; 1])/\mathcal{T}$-measurable functions $\mathcal{D} := \{d | d : [0; 1] \to T\}$.

Definition 5.2.1. If the mobility model f_m is a deterministic function, then we will call the following functions - layer functions. Each is defined in its proper, measurable spaces. Note that these functions all depend on the specific choice of the zones $\mathcal{Z}$, which we will omit for brevity.

(i) A strategy layer function is:

$$l_{str} : \Omega_{str} \times \mathbb{N} \to \mathcal{Z} \times T \tag{5.3}$$

(ii) A mapper layer function is:

$$l_{map} : \Omega_{map} \times \mathbb{N} \times \mathcal{Z} \to \mathfrak{Z}, \tag{5.4}$$

(iii) A tactic layer function is:

$$l_{tac} : \Omega_{tac} \times \mathbb{N} \times \mathfrak{Z} \times \mathfrak{Z} \to \mathcal{G} \tag{5.5}$$

where $\forall(\omega, n, e_{src}, e_{dst}) \in \Omega_{tac} \times \mathbb{N} \times \mathfrak{Z} \times \mathfrak{Z}$, if we denote $g = l_{tac}(\omega, n, e_{src}, e_{dst})$, then $g(0) = e_{str}$ and $g(1) = e_{dst}$.

(iv) A dynamic layer function is:

$$l_{dyn} : \Omega_{dyn} \times \mathbb{N} \times \mathcal{G} \to \mathcal{G} \times \mathcal{D} \tag{5.6}$$

(v) A stay layer function is:

$$l_{sta} : \Omega_{sta} \times \mathbb{N} \times T \times \mathcal{Z} \to \mathcal{G} \times \mathcal{D} \tag{5.7}$$

where $\forall(\omega, n, t_{stay}, Z) \in \Omega_{sta} \times \mathbb{N} \times T \times \mathcal{Z}$, if we denote $(g, d) = l_{sta}(\omega, n, t_{stay}, Z)$, then $g(0) \in Z$ and $g(1) \in Z$.

The definition of these functions is with direct correspondence to the layers defined by LEMMA. All layer functions are defined on their corresponding measurable parameter spaces $(\Omega_{idx}, \mathcal{F}_{idx})$ and the given path segment number $n \in \mathbb{N}$. We are not going to mention these two variables in the justification why they are present in the layer function definitions.

The strategy layer determines the destination zone and the stay time, which is the same as the layer function l_{str}, which returns the next zone to be visited ($Z_{dest} \in \mathcal{Z}$) and the stay duration ($t_{stay} \in T$). It is worth mentioning, that in this case we can separate the function l_{str} into two separate functions, one for the destination zone, and one for the stay time.

The mapper layer selects a point from the destination zone. The layer function picks the destination point ($e_{dest} \in E$) from the given destination zone ($Z_{dest} \in \mathcal{Z}$).

The tactic layer chooses the route to be taken between two points. The corresponding tactic layer function takes the source and destination points ($e_{src}, e_{dst} \in \mathfrak{Z}$) and returns the route to be taken $g \in \mathcal{G}$. Note that $g(0) = e_{src}$ and $g(1) = e_{dst}$.

The dynamic layer adds the speed to the trajectory, as well as (possibly) small deviations to it. So, the dynamic layer function projects the selected route $g \in G$ to (possibly the same) a trajectory $g_1 \in G$ and the time at which the node should move $d \in D$. In fact, the movement of the node in the given segment is specified by $g(d^{-1}(t))$. Note that d^{-1} exists because d is monotone increasing function.

The stay layer determines the movement of the layer in the selected destination zone for the stay duration. The stay layer function projects the stay duration ($t_{stay} \in T$) and the destination zone ($Z_{dest} \in \mathcal{Z}$) into the movement pattern to be followed by the node ($(g, d) \in \mathcal{G} \times D$).

Theorem 5.2.2. *The layer functions as defined in 5.2.1 specify a mobility model as formalized in 5.1.2.*

In order to prove the theorem, we will use a helper lemma (the proof can be found in A.4.1):

Lemma 5.2.3. *A mobility model as defined in 5.1.2 can be constructed with the help of two families of functions -* $\Gamma := \{\gamma_i | \gamma_i : \Omega \times E \to \mathcal{G}, i \in \mathbb{N}_+\}$ *and* $\Delta := \{\delta_i | \delta_i : \Omega \times E \to \mathcal{D}, i \in \mathbb{N}_+\}$ *all defined over a common sample space* Ω*, provided that all functions in* Γ *and* Δ *are measurable.*

For the full proof of the theorem, please refer to A.4.2.

5.3 PROBABILISTIC MODELS

Definition 5.3.1. If the mobility model is a stochastic process, then we will call the processes - layer processes, each is defined in its proper, probability space. As in the processes depend on the specific choice of the zones $\mathcal{Z}$, which we will omit for brevity.

(i) A strategy layer process is:

$$\mathbf{X}_{str} = \{X_n | X_n : \Omega_{str} \to \mathcal{Z} \times T, n \in \mathbb{N}\} \tag{5.8}$$

(ii) A mapper layer process is:

$$\mathbf{X}_{map} = \{X_n | X_n : \Omega_{map} \times \mathcal{Z} \to \mathfrak{Z}, n \in \mathbb{N}\} \tag{5.9}$$

(iii) A tactic layer process is:

$$\mathbf{X}_{tac} = \{X_n | X_n : \Omega_{tac} \times \mathfrak{Z} \times \mathfrak{Z} \to \mathcal{G}, n \in \mathbb{N}\} \tag{5.10}$$

(iv) A dynamic layer process is:

$$\mathbf{X}_{dyn} = \{X_n | X_n : \Omega_{dyn} \times \mathcal{G} \to \mathcal{G} \times \mathcal{D}, n \in \mathbb{N}\} \tag{5.11}$$

(v) A stay layer process is:

$$\mathbf{X}_{sta} = \{X_n | X_n : \Omega_{sta} \times T \times \mathcal{Z} \times \mathcal{G} \to \mathcal{G} \times \mathcal{D}, n \in \mathbb{N}\} \tag{5.12}$$

Theorem 5.3.2. *The layer stochastic processes as defined in 5.3.1 specify a mobility model as formalized in 5.1.3.*

Proof. The proof follows from the observation that we can represent the sample paths of the layer processes as their respective layer functions (defined in 5.2.1). Afterwards, by applying Theorem 5.2.2 we obtain the sample paths of the stochastic process of the mobility model. □

5.4 LEMMA REPRESENTATIONS

5.4.1 *Trivial representations*

Here, we are going to give two trivial LEMMA representations of a mobility model. They are called trivial because they require very specific constraints on the environment components. Furthermore only one of the layers is carrier of the core movement principles, and that layer cannot function in any other setup (such as having two zones non-covering the simulation area).

5.4.1.1 *One-zone trivial representation.*

If we consider the following setup: a simulation with a single zone covering the entire simulation area, with no constraints nor movement influencing factors. The strategy always selects the single existing zone, the tactic performs linear motion with a constant movement time dynamic, and no stay layer. The constant movement time dynamic always finishes the movement segment for a fixed period of time (for example 1 s.). Given that setup, the mapper completely specifies the node movement. Any model may be plugged as a mapper, with the movement being sampled.

5.4.1.2 *All-zones trivial representation.*

Let us consider the setup where a zone is created for each point of the simulation area. The mapper selects the given point from the zone, the tactic is constant movement, with the constant time dynamic. Any model may be used as a strategy layer, thus producing an arbitrarily close approximation of the original movements.

5.4.2 *Nontrivial representations*

The one-zone and all-zones trivial representations of an arbitrary mobility model are certainly very useful, but they impose limitations on LEMMA's universal characteristics. Here we are going to state a much stronger claim, namely:

Theorem 5.4.1. *There exists at least one non-trivial LEMMA representation for each mobility model* m.

Please refer to A.4.3 for the proof of the theorem.

As an illustration, let's suppose we want to find the LMR of a given sample path, as the one shown in Fig. 44. To facilitate the visualization, we have chosen a 1D simulation area with two zones. The strategy layer provides the zones to be visited (here $Z_2, Z_1, Z_1, Z_2, Z_2, Z_1$) along with the stay times, the mapper select a point of the zone (here designated with white dots), the tactic layer chooses the path to be followed and the dynamic layer generates the final curve (the segments starting with a black and ending with white dots). Afterwards, the stay layer generates the movement in a zone (the segments starting with white and ending with black dots).

Corollary 5.4.2. *There exist more than one non-trivial LEMMA representation for each mobility model* m *provided* m *and the environment allow more than one representation.*

Proof. If the mobility model and the environment are not pathological cases (such as having a simulation area consisting of a single point) the proof of Theorem 5.4.1 may be modified in such way, as to select different point from the destination zones, thus modifying the different layers and providing an alternative LMR representation of the model. □

5.5 STATIONARITY

It is important to know if a model is terminating or stationary, as its type affects the simulation results and if handled improperly can lead

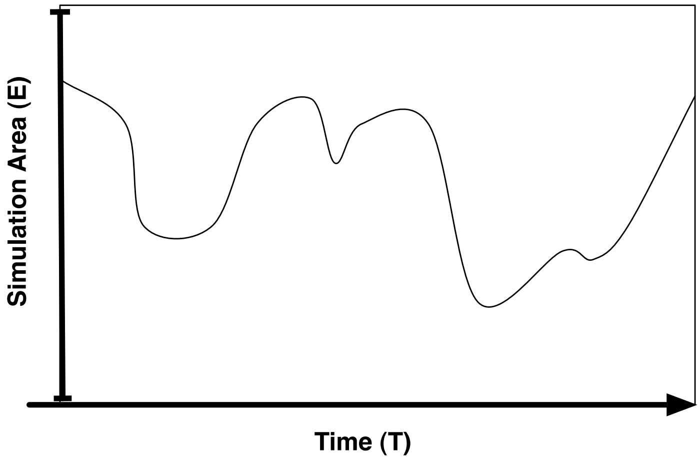


Figure 44: A sample path of a 1D mobility model.

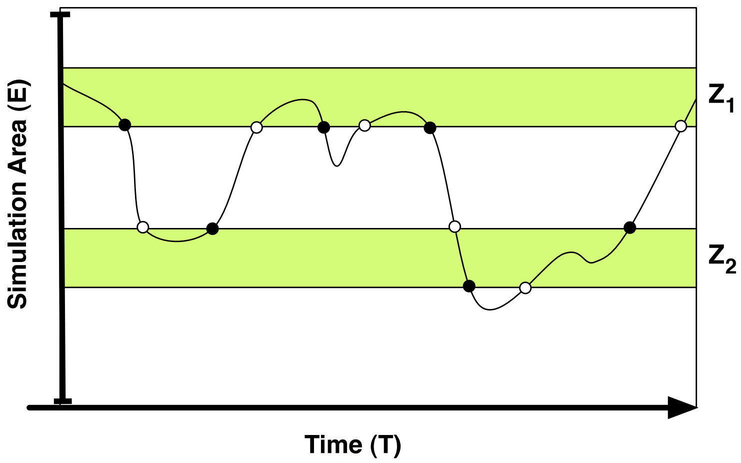


Figure 45: LEMMA representation of the sample path shown in Fig 44. Z_1 and Z_2 are the zones defined for this scenario.

to biased conclusions. A terminating simulation is executed until a particular event occurs, upon which the simulation is stopped. A non-terminating simulation can have an arbitrary duration, during which the behavior of the entities should follow some kind of regular behavior. In all cases, node mobility should not affect the performance of the simulated network in unexpected ways. An important characteristic of the mobility models is thus whether if they possess a stationary regime. Here, we are going to provide a way to construct stationary mobility models out of independent layers, where stationary process is a stochastic process with probability laws that do not change over time.

Theorem 5.5.1. *A probabilistic mobility model constructed in the form specified by LEMMA (excluding the stay layer) is stationary if its layers are all stationary, pairwise independent processes. Additionally, all curves in $\mathcal{G}$ and $\mathcal{D}$ should be with a finite length.*

Note that this is a sufficient, but not a necessary condition, e.g. it may be possible to construct a stationary mobility model even though some of its layers may be non-stationary processes.

The proof is provided in A.4.4, and uses the following lemmas:

Lemma 5.5.2. *If $X : \Omega \to F^{\mathbb{N}}$ and $Y : \Omega \to B^{\mathbb{N}}$ are pairwise independent, stationary processes (defined in appropriate spaces), and $F \subseteq A^B$ are measurable functions, then $V := f_2(X, Y)$, $V : \Omega \to A^N$ is a stationary process.*

Proof. By definition $V(\omega, n) = f_2(X(\omega, n), Y(\omega, n))$, where $X(\omega, n)$ is a measurable function such that $X(\omega, n) : B \to A$. The random process (X, Y) of two pairwise independent, stochastic random variables is also stationary. Finally, f_2 is measurable (F are measurable functions, and X and Y are measurable random functions) which by *Lemma 9.1* from [Kal02] provides the stationarity of V. □

Lemma 5.5.3. *If $X : \Omega \to F^{\mathbb{N}}$ and $Y : \Omega \to B^{\mathbb{N}}$ are pairwise independent, stationary processes (defined in appropriate spaces), $F \subseteq A^{B \times B}$ are measurable functions, and θ a measure-preserving shift transformation, then $W := f_3(X, Y, \theta)$, $W : \Omega \to A^N$ is a stationary process.*

Proof. As θ is a measure preserving transformation, θY is a stationary process, which analogically to 5.5.2 provides the measurability of f_3, and thus, the stationarity of W. □

Corollary 5.5.4. *A probabilistic mobility model constructed by the form specified by LEMMA (excluding the stay layer) is stationary if all curves in $\mathcal{G}$ and $\mathcal{D}$ are with finite length and one of the two conditions is fulfilled:*

1. *Both L'_{tms} and L'_{dyn} are stationary, pairwise independent processes.*
2. *L'_{ms}, L'_{tac} and L'_{dyn} are stationary, pairwise independent processes.*

Proof. The result follows directly from Theorem 5.5.1 and just gives the precision that an appropriate combination of non-stationary layer processes may be used in conjunction to construct a stationary process. □

Theorem 5.5.5. *A process specified as described in Theorem 5.5.1 with a pause stay layer is stationary.*

Proof. Adding a pause stay layer (e.g. there is no movement specified during the move duration) to a model m, specified with the layers processes L_{str}, L_{map}, L_{tac} and L_{dyn} makes the process equivalent to a process m^1 specified with the layers L_{str}, L_{map}, L_{tac} and L^1_{dyn}, where L^1_{dyn} is a modification of L_{dyn} which given the stay time t_s (generated by the stationary and independent process $L_{str,t}$) transforms the pair $(g, d) \in \mathcal{G} \times \mathcal{D}$ to $(g^1, d^1) \in \mathcal{G} \times \mathcal{D}$ as follows:

$$g^1(x) := \begin{cases} g(2x) & x \in [0; \frac{1}{2}] \\ g(1) & x \in (\frac{1}{2}; 1] \end{cases}$$

$$d^1(x) := \begin{cases} d(2x) & x \in [0; \frac{1}{2}] \\ d(1) + (2x - 1)t_s & x \in (\frac{1}{2}; 1] \end{cases}$$

Indeed, the modification applied to the dynamic layer prolongs each movement cycle with the pause duration, and the trajectory is modified, so that there is no movement during the pause time. This dynamic layer is indeed stationary, as both the original dynamic layer, and the strategy layer (specifying the pause duration) are pairwise independent stationary processes, and the transformation $(g, d) \to (g^1, d^1)$ is measurable. Thus, from Theorem 5.5.1 m^1 is stationary, which as we noted is the equivalent of m with pauses. □

Note that this construction is also applicable to a more general case, where the stay layer process is a limited movement around the endpoint of the node, the extent of these limits being dependent on the simulation environment.

5.5.1 *Simulation*

In order to show experimentally the results from the previous section, we have performed a series of simulations. We have selected stationary layer processes for 5 strategies, 5 mappers, 4 tactics and 3 dynamics, summarized in Table 2, and have evaluated the stationarity measures of all possible resulting mobility models (300 models). They were evaluated in nine different setups in a 1D environment shown in Fig. 5.5.1 - four patterns for the 1000 m simulation area, five patterns for the 2000 m area. The average node density was fixed to 1/10 nodes/meter, which gives 100 nodes for the 1000 m simulations and 200 nodes for the 2000 m simulations.

Each of the simulations was 22000 s long and was ran 10 times, each time with a different random generator seed (each experiment had a different seed). The collected data were then treated as time series. The nodes were initially placed uniformly on the simulation area, and first 2000 s of the simulation were discarded to remove any initialization bias.

The analyzed data include the position and the speed of each node, as well as the distribution of the nodes over the simulation area, and the distribution of their speeds. All data were sampled at 1 second intervals, and were then submitted to the augmented Dickey-Fuller test (ADF) [SD84, BDGH93]. The ADF tests the data against the null hypothesis of a unit disk root, with alternative hypothesis of stationarity of the data. Thus, a sufficient condition for a (sufficiently long) sample to be coming from a stationary process is the null hypothesis to be rejected.

Table 2: Name and description of the different layers used in the simulations.

Strategies	
Constant	Always select a fixed zone.
Enumerated	Select the first zone, then the second, until reaching the last zone, then restart from the first.
Uniform	Select the zone with probability of a uniform distribution.
Normal	Select the zone with probability of a normal distribution.
Exponential	Select the zone with probability of an exponential distribution.
Mappers	
Midpoint	Select the center of the zone.
Upper point	Select the maximal point of the zone.
Lower point	Select the minimal point of the zone .
Uniform	Uniformly select the point of the zone.
Normal	Select the point with normal distribution.
Tactics	
Linear	Move linearly towards the destination.
Tango	Move 2/3 of the way to the destination, then step back 1/3 of the way, then go to the destination point.
Reversed Tango	Move 1/3 away from the destination, then move $1\frac{1}{3}$ in the direction of the destination (by surpassing it), and then step back with 1/3 of the distance, in order to reach the endpoint.
Random Tango	With a probability p move with Tango, and probability $1-p$ with Reversed Tango.
Dynamics	
Constant	Move at a constant speed.
Uniform	Draw the speed from a uniform distribution and move on the entire trajectory with that speed.
Exponentially Spaced	Draw the speed from a uniform distribution and move for a duration drawn from an exponential distribution.

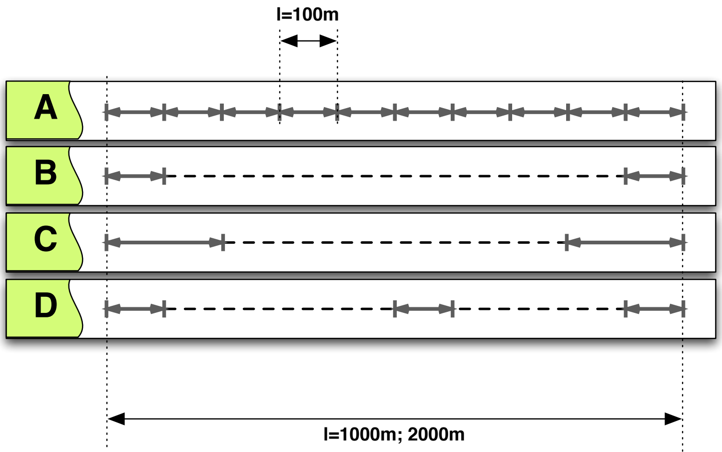


Figure 46: The different environments used for the simulations. All four types of zone dispositions have been used with the 1000m and 2000m simulation areas. For the 2000m area we have also performed simulations with zones with length 200m following pattern D.

Table 3: Percent of traces per trace type for which the ADF test result was statistically significant at the 1% level

Node position	77.5%
Node speed	100%
Node sojourn distribution	100%
Node speed distribution	97.9%

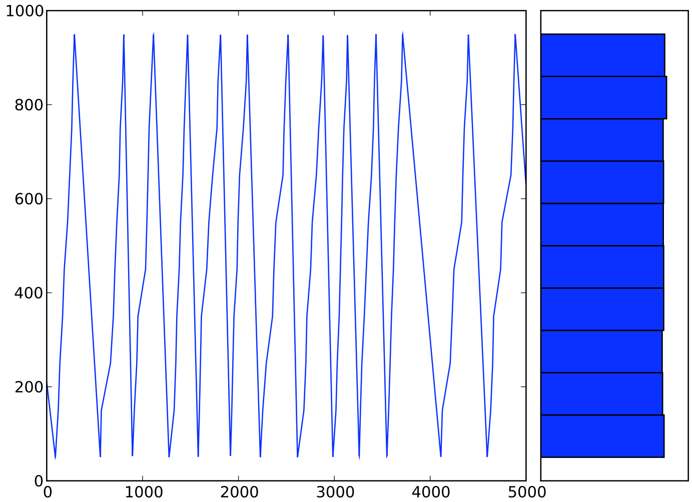
5.5.2 *Results*

All 1D movement traces without exception were recognized by the ADF test as stationary time series (e.g. the null hypothesis was rejected). Moreover, the majority of the results obtained from the test are statistically significant at the 1% level (as summarized in Table 5.5.2). In Figures 47 and 48 you can observe sample realizations of various mobility models used in the analysis, along with the node sojourn density.

5.6 CONCLUSION

This chapter was aimed at the definition and the study of the mathematical properties of our architecture - LEMMA - introduced in the previous chapter (Chapter 4). Here, we provided the fundamental results necessary for LEMMA to be used as a universal mobility model framework both for empirical and for theoretical models.

We gave a formal definition of mobility model for the deterministic and probabilistic cases. The definition is very general and encompasses continuous- and discrete- time models. Furthermore, we formalized the layer functions and processes corresponding to the layers as defined in Chapter 4.



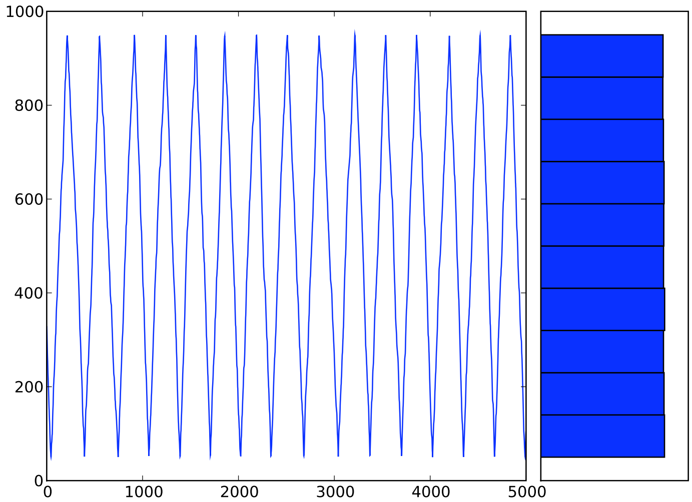


Figure 47: Setup - 100 nodes, 1000 m area, environment A. Sample realization of 5000 s of a LEMMA process and the corresponding aggregated node sojourn distribution for: (top) Enumerated strategy, Midpoint mapper, Linear tactic and Uniform dynamic, and (bottom) Enumerated strategy, Midpoint mapper, Linear tactic and Exponentially Spaced dynamic.

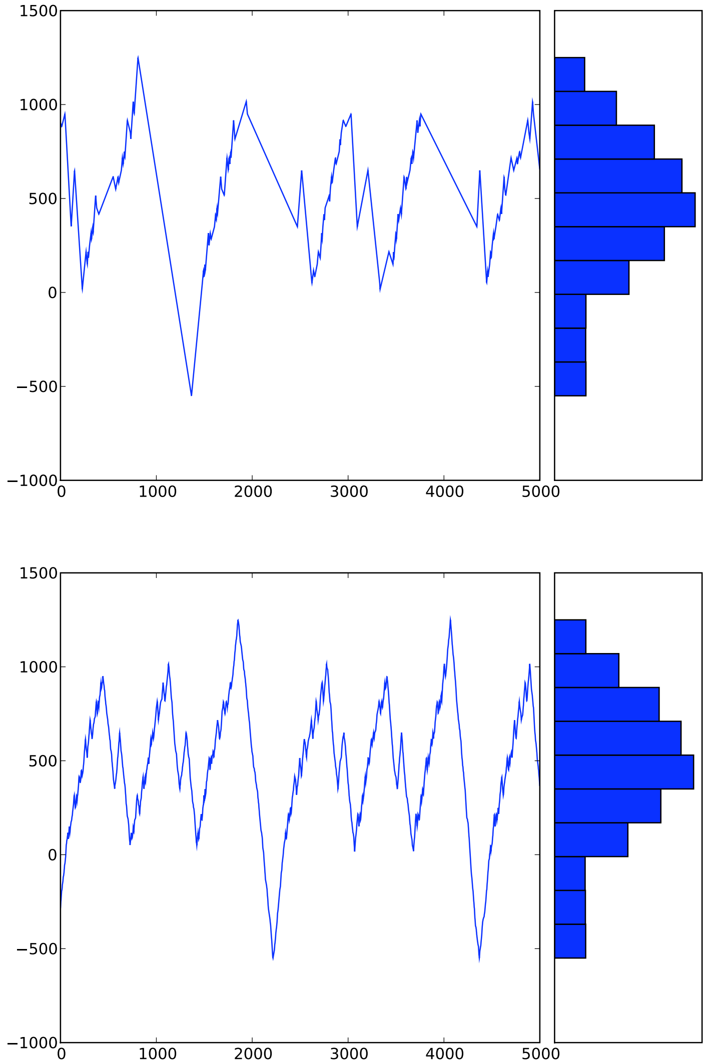


Figure 48: Setup - 100 nodes, 2000 m area, environment A. Sample realization of 5000 s of a LEMMA process and the corresponding aggregated node sojourn distribution for: (top) Enumerated strategy, Midpoint mapper, Random Tango tactic and Uniform dynamic, and (bottom) Enumerated strategy, Midpoint mapper, Random Tango tactic and Exponentially Spaced dynamic.

The first important result is the fact that any combination of valid layer functions (Theorem 5.2.2) or layer processes (Theorem 5.3.2) defines a valid mobility model. This shows that there are no inconsistencies in the way the layers and the layer functions and processes were defined. It also provides the guarantee that once defined, a given layer can be used with any other combination of layers without the risk of incompatibilities.

Any mobility model can be represented with LEMMA in a somewhat naive, or otherwise said "trivial" way, which does not allow layer generalization and requires specific environment. The second important result is Theorem 5.4.1, which states that any mobility model has at least one non-trivial LEMMA representation. This means that *any* mobility model may be represented as a collection of independent layers, which can also be recombined with other layers, and be used in different types of environments. This theorem is the major result of this chapter, as it provides a formal proof of the claim we already supported with empirical observations (see Section A.1), namely that LEMMA can be used as a universal mobility model framework.

Finally, we have shown that it is possible to obtain global model knowledge by only having the properties of the individual layers. As an example, we have investigated a fundamental stochastic process property - process stationarity. In Theorems 5.5.1 and 5.5.5 we proved that the stationarity of the used layer processes guarantees stationarity of the model they induce, provided they are pairwise independent. Finally, we have supported these findings by simulating 300 different mobility models in 9 environmental setups.

After the definition of LEMMA in the previous chapter, we have provided its mathematical foundations along with several important theorems. In the next chapter (Chapter 6) we provide application for the architecture by providing layers based on several real-world datasets. Mobility models built with these layers are then validated in contexts extracted from the same datasets.

Part III

VALIDITY CONTEXT DEFINITION

6 GPS-TRACE BASED VALIDITY CONTEXT DEFINITION

In this chapter, we provide an example of the definition of context of validity. We have analyzed two datasets of GPS traces gathered by taxi fleets in Nagoya, Japan and San Francisco, USA. We use these datasets to determine the contexts of validity (as defined in Section 2).

6.1 DATA DESCRIPTION

6.1.1 *San Francisco Taxi Fleet Data*

The data is gathered by 536 taxis and covers the period from 17 May 2008 to 10 June 2008 for a little more than 3 weeks (24 days). It has been compiled and studied by Piorkowski et al. [PSDG08] based on the CabSpotting project [cab] and made available for the public data repository *CRAWDAD* [PSDG09a]. The active taxis per day are shown in Fig. 49. The first and last days of the trace contain data for the second and first half of the day, respectively. We have studied the period from 18 May 2008 to 6 June, or exactly 3 weeks of data.

6.1.2 *Nagoya Taxi Fleet Data*

The Nagoya traces have been obtained as part of our collaboration with the University of Kyoto, Japan. The data are gathered by 1226 taxis and cover the period from 1 October 2002 to 16 March 2003. They were collected as part of some traffic engineering efforts, which were focused on street detour selection [KYYM03], route selection and toll system efficacy and appreciation [MM03] and meteorologic conditions effects on speed [WYMM06]. We are using the same dataset, aimed at modeling characteristics relevant to wireless network researchers. There are two types of trace gathering mechanism used in this study, which depend on the installed equipment in the taxis. A total of 764 type 1 taxis, and 472 type 2 taxis were recorded (10 of the cars were upgraded from type 1 to type 2 during the study).

As it can be seen in Fig. 50, in the first part of the given period only type 1 taxis were recorded, while type 2 taxis are recorded only after 14 December 2003. Additionally, there are significant fluctuations in the number of taxis around the holidays. We therefore concentrated our efforts on analyzing a period of 4 weeks during which the number of active taxis is consistent - from 1 February 2003 to 1 March 2003.

6.2 DATA PREPARATION

6.2.1 *GPS data treatment*

The GPS location of each taxi is recorded at fixed intervals or upon the occurrence of a certain type of event (e.g. sudden stop). Typically, a taxi is logged at a granularity which can vary from several seconds to 5

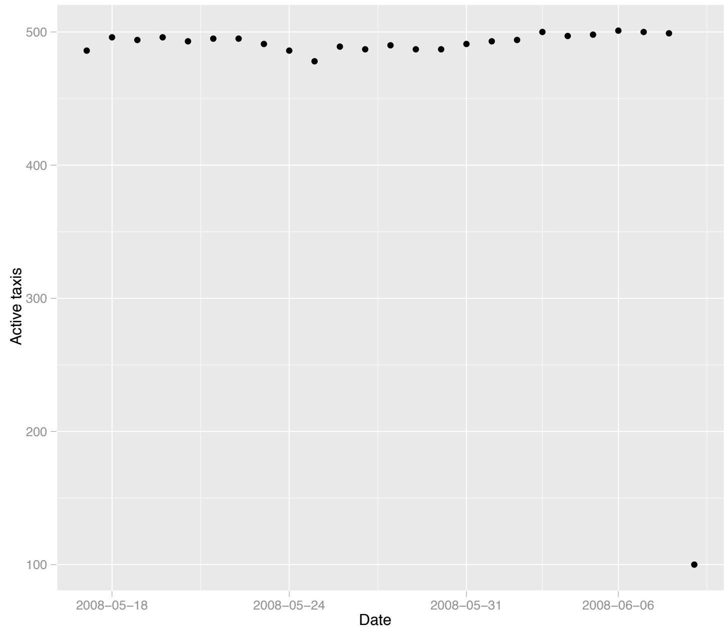


Figure 49: San Francisco dataset: Number of unique taxis active per day. The studied period is from 18 May 2008 to 6 June 2008.

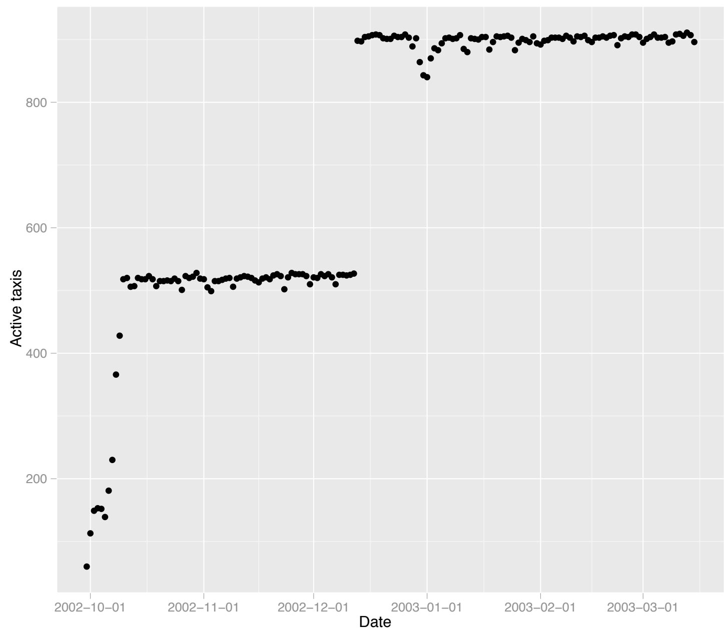


Figure 50: Nagoya dataset: Number of unique taxis active per day. The studied period is from 1 February 2003 to 1 March 2003.

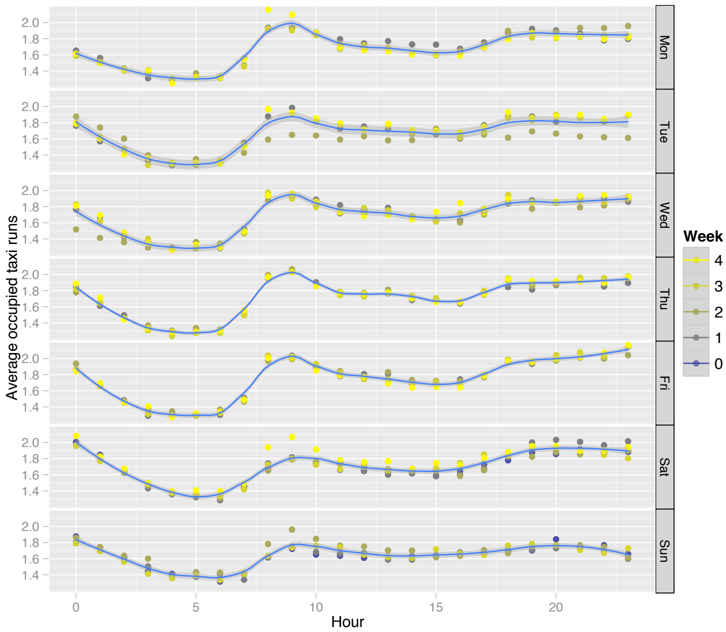


Figure 51: Nagoya dataset: Number of average runs per taxi, grouped by hour of day, day of week and week of the trace.

minutes, depending on the type of traces and used trace mechanism. In order to obtain the intermediate positions of the taxis, we performed a linear interpolation with granularity of 1s between the recorded locations. This process, together with the possible inaccuracies of the locations estimated with GPS make the instantaneous position of an individual taxi unreliable. However, because we are studying the movements in the context of wireless networks, small deviations can be tolerated. Moreover, due to the significant volume of available data, the general movement characteristics of the entire set of taxis remain unaffected. We have studied the contacts between taxis, where two taxis are considered to be in contact when the distance between them is equal or less than 150m. The distance has been selected after ensuring that the structure of the properties we have studied do not change under shorter/longer ranges (50m, 100m, 300m and 600m) and is consistent with the transmission ranges in WiFi networks.

Every taxi reports its position, together with the time instant at which the position was recorded, and the state of the taxi - if it is free, or occupied. We divided the movement of the taxis into taxi "runs", where a run is a movement period during which a taxi is servicing a client or during which it is empty. The San Francisco dataset contains a total of 4464385 runs, of which 1999692 were when servicing a client, and 2464693 when the taxi was free. The Nagoya dataset contained 4318590 runs - 2288344 occupied and 2030246 free.

6.2.2 *Taxi clustering*

We have calculated the number of encounters between taxis and have discovered that they do not form a homogeneous group. Instead, the taxis seem to follow different types of movement patterns, which may

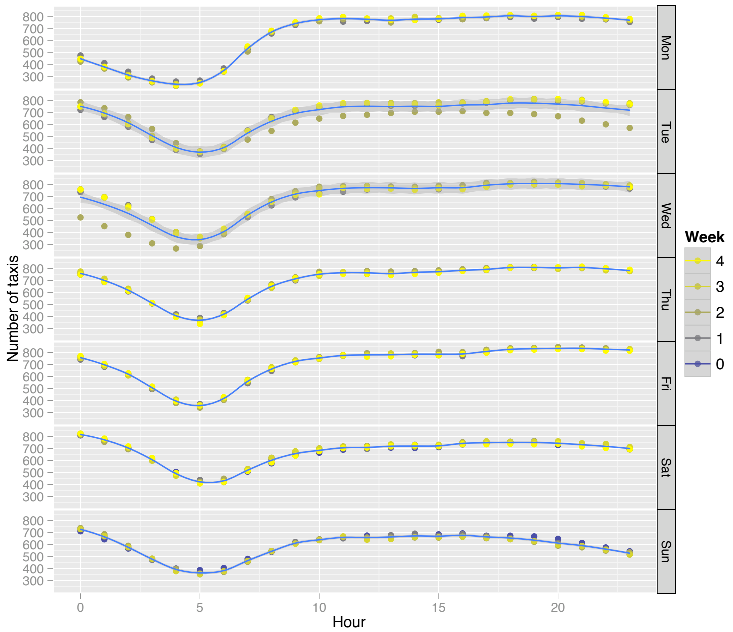


Figure 52: Nagoya dataset: Number of active taxis per hour, day of the week and week of the trace.

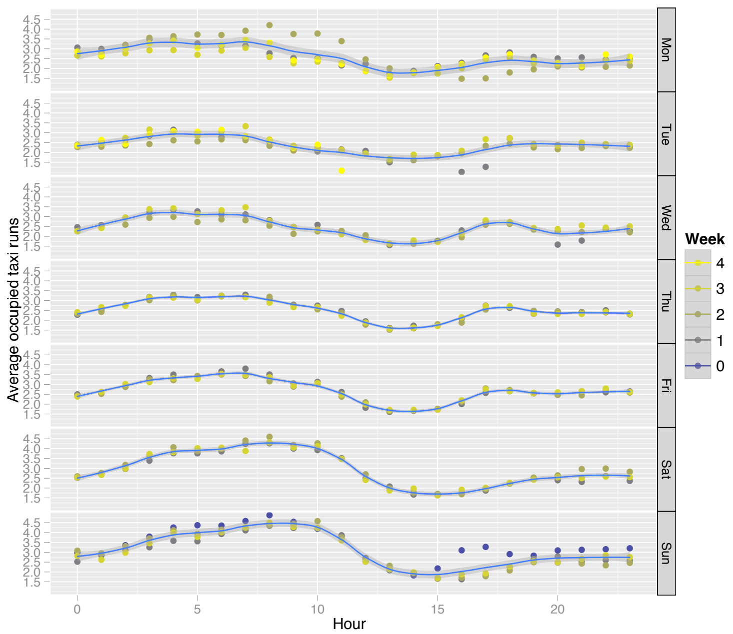


Figure 53: San Francisco dataset: Number of average runs per taxi, grouped by hour of day, day of week and week of the trace.

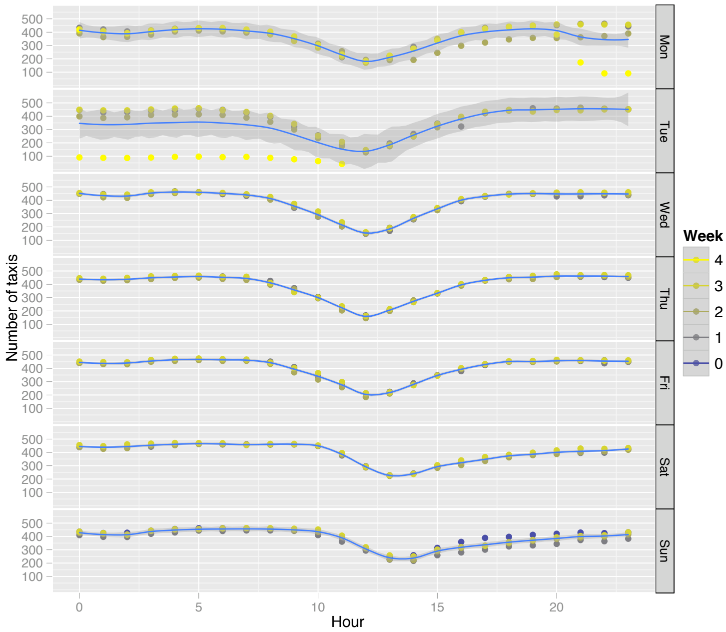


Figure 54: San Francisco dataset: Number of active taxis per hour, day of the week and week of the trace.

be conditioned on spatial and/or temporal factors (for example, in the Nagoya dataset, there is a group of 20 taxis which are active only during the weekends). In order to deal with compatible underlaying mobility principles we wanted to categorize the similar taxis and perform the further analyses on a per-group basis. For that end, we have used X-means [PM00] as clustering algorithm, which is a modification of the K-means algorithm, with automatic estimation of k. The method uses Bayesian Information Criterion (BIC) to compare the goodness of fit of the different matching models in order to estimate the parameter k. For each taxi i, we calculated its d-dimensional encounter vector (where d is the number of taxis in the trace), whose j-th component holds the number of encounters with the j-th taxi. This provides us with the $d \times d$ taxi encounter matrix E whose elements $e_{i,j}$ hold the number of encounters between the i-th and the j-th taxis. Finally, the Nagoya taxis were divided into 6 clusters, and the San Francisco taxis were divided into 5 clusters.

Figures 56 and 56 show the encounter matrix of the Nagoya and San Francisco taxis respectively. The upper matrices show the default ordering of the taxis, while the lower illustrates the encounters after sorting the taxis by their cluster.

6.2.3 *Time period separation*

Taxi movement patterns follow the rhythm of travel demand, and consequently, manifest time-varying characteristics. As an example, the number of taxi encounters (Fig. 57 and 58) varies greatly depending on the hour and the day of week, with weekdays having much more encounters than weekends, and day-time periods being less active than evening and morning hours. We have illustrated the number of

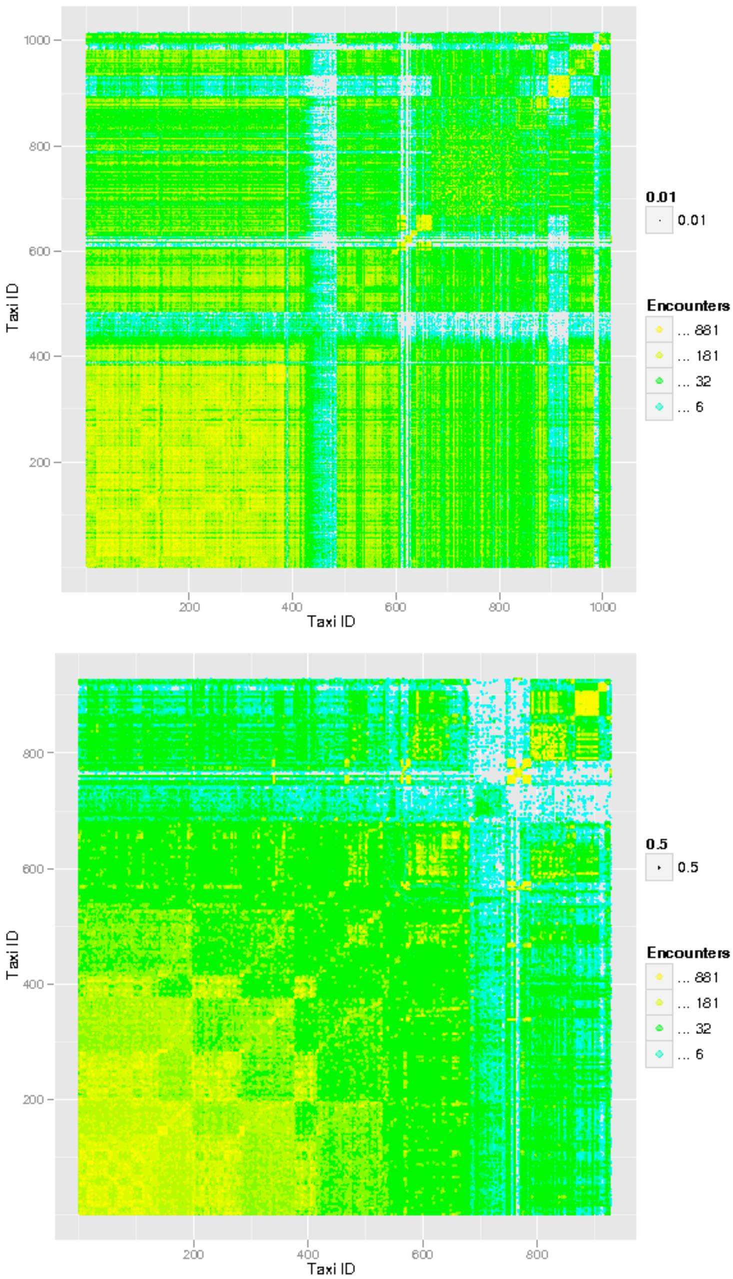


Figure 55: Nagoya dataset: Encounters between taxis: top - unordered, bottom - ordered by their clusters.

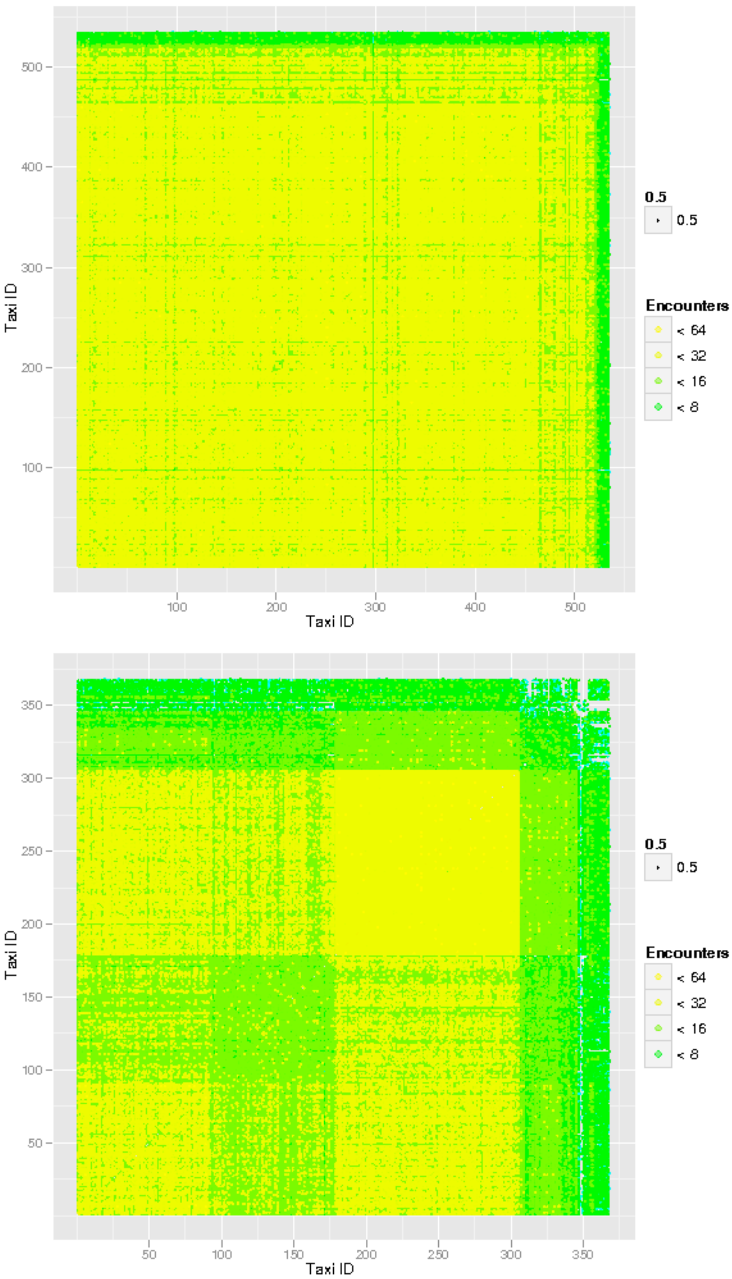


Figure 56: San Francisco dataset: Encounters between taxis: top - unordered, bottom - ordered by their clusters.

encounters, as well as the logarithm of the number of encounters in order to better illustrate the difference in the patterns of Nagoya and San Francisco. The Nagoya dataset has its lowest time of usage in the period from 3 to 8 am, having the peak usage from 17 h onwards. San Francisco taxis have their lowest encounter rate in the afternoon hours (11-15h), peaking around midnight and with much less overall difference.

In order to treat homogeneous movement patterns, we have divided the trace into weekdays and weekends, as they are a source of significant variance in the patterns of both datasets. Furthermore, after analyzing the number of runs per hour (Fig. 51 and 53), and after clustering the day hours on several movement-related criteria (e.g. number of runs from/to a zone per hour) we decided to split each day into four periods - from 3 h to 5 h, from 6 h to 11h, from 12 h to 20 h, and from 21 h to 2 h. This division is mostly based on the strong variance of the patterns in the Nagoya dataset, but we've applied the same division to the San Francisco dataset for symmetry.

6.3 CONTEXT OF VALIDITY

As defined in Section 2, in order to validate a mobility model we need to define the context in which it will be considered valid. The metrics we are going to use for the validation of our model are based on some of the most frequently used ones:

- Aggregated Inter-Contact Time (ICT) distribution
- Pairwise Inter-Contact Time (ICT) distribution
- Aggregated CD distribution
- Pairwise CD distribution

The Inter-Contact Time (ICT) as defined earlier, is an important metric that has been used to validate several existing mobility models, e.g. [BLDAF09, LHK+09]. Here, we are going to distinguish the Aggregated ICT which is the distribution taken over all nodes from the Pairwise ICT, which represents the distribution of ICT between a single node and the rest of the population. Indeed, studying the latter case shows that the movement patterns of the nodes cannot be summarized only with the aggregated metric, a question already discussed in [CLF07]. In addition, we studied the Contact Duration (CD) distributions. We chose these metrics, because they play an important role in the characterization of the performance of DTN networks and the type of movement patterns involved. Most importantly, these metrics seem to be of the few widely studied and accepted movement measurements (e.g. applied to university WLAN [HH05], Bluetooth [CHD+07, SCP+04] and GPS traces [LHK+09]) which are applicable to human and vehicular movements.

From the existing studies of ICT and CD distributions, it was shown that they typically follow a heavy-tailed distribution. We have fitted distributions which have been shown to model these interactions in various contexts - Pareto (power law), power law with exponential cutoff, exponential and log-normal. We have used Maximum Likelihood Estimation (MLE) to estimate the values of each of these parameters as described in [CSN07]. We have then estimated the goodness of

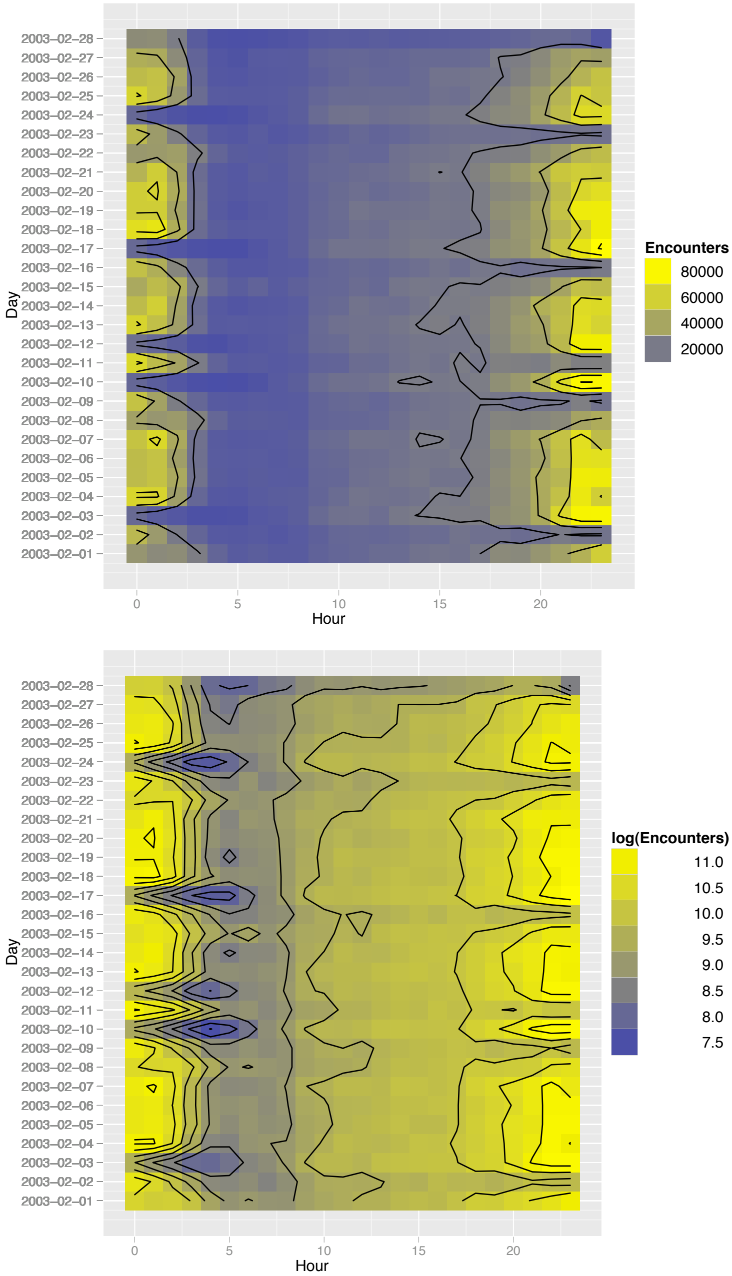
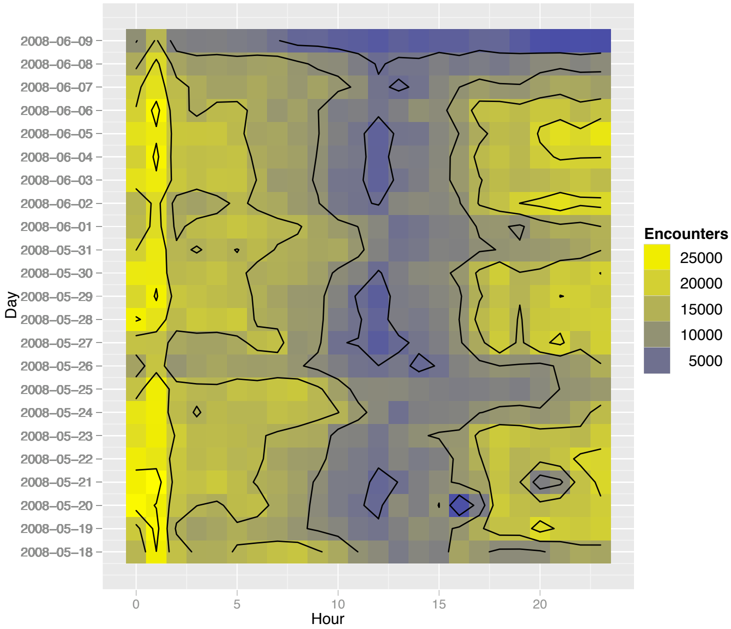


Figure 57: Nagoya dataset: Number of encounters per hour and per day of week. The lower figure illustrates the $\log_2$ of the number of encounters.



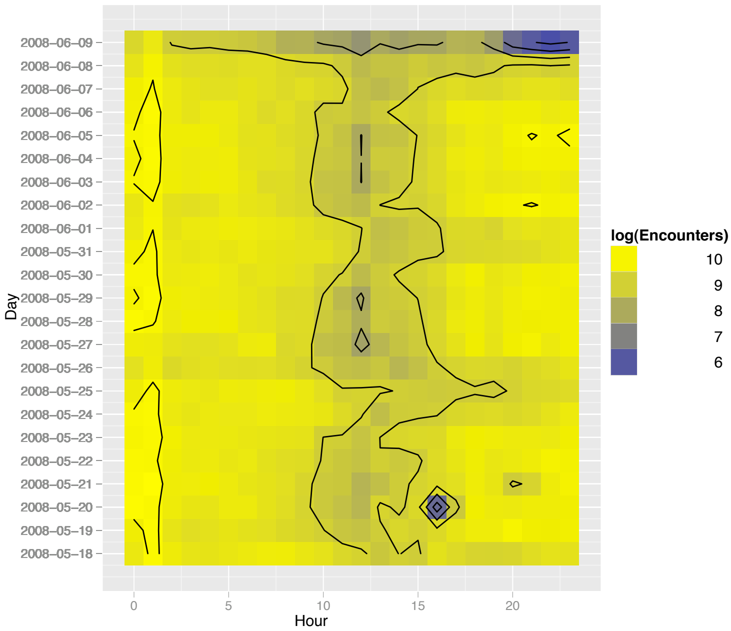


Figure 58: San Francisco dataset: Number of encounters per hour and per day of week. The lower figure illustrates the $\log_2$ of the number of encounters.

fit of each of these models by calculating the Kolmogorov-Smirnov statistic and then calculating the corresponding p-value with the help of Monte-Carlo simulations (as described in [CSN07]).

Additionally, we have used the log-likelihood ratio to select the best fitting model. We have compared all pairs of models, where the sign of the logarithm of the ratio of the likelihoods indicates which of the two is a better fit, and is zero if they are tied. In order to estimate how significant this difference is, we have used the method proposed by Vuong [Vuo89].

We have performed our calculations for each minute of trace data. In order a model to be considered valid in the context of Nagoya or San Francisco taxi movements, its Aggregated ICT and CD distribution must be the same as the ones discovered in the data, with close parameters. Additionally, the Pairwise distributions should have the same proportions of matching distributions, with close parameter values.

6.3.1 *Nagoya*

The aggregated ICT and CD are shown in Fig. 59 and 60. Example pairwise distributions are given in Fig. 61 and 62.

For the pairwise ICT distribution, we have found that the best matching model is the power law with exponential cutoff, which matched with $p - value = 0.99$[1] all taxis of the trace. The second best model turned out to be the log-normal distribution, with a $p - value = 0.81$ for all but 4 taxis, which were matched with $p - values$ of $0.6 - 0.65$ and a single taxi with $p - value = 0.2$. This means that the log-normal distribution fits well the data and is a valid (albeit slightly worse) alternative to the power law with cutoff. The pairwise ICT of none of the taxis were fitted well with the Pareto distribution, and only one taxi was slightly estimated by the exponential distribution (with $p - value = 0.21$). These findings are additionally confirmed from the log-likelihood ratio test, where the power law with exponential cutoff fits well the entire population of taxis, being tie with the log-normal distribution in 3 cases.

The parameters of the power law with exponential cutoff are its exponent and rate. They are spread with the exponent=0.617952 ± 0.30229 and rate=0.000211 ± 0.000241. The parameters of the log-normal distribution on the other hand are in a much tighter interval - logarithmic mean=6.317746 ± 0.598991 and standard deviation=2.038956 ± 0.222208. This could mean that even though the power law with cutoff distribution is a better fit than the log-normal, the latter may be used as additional validation metric, thanks to the closeness of its parameters.

It is important to note that the parameters of the aggregated ICT are very close to the averaged values of the pairwise ICT - for the power law with exponential cutoff the exponent is 0.6341496, and the rate is 0.0001926736, while for the log-normal distribution the mean-log is 6.278567 and the sd-log is 1.996686 e.g. the aggregated parameters may be used in rough approximation and validation procedures. As with the pairwise ICT distribution, the log-likelihood ratio favors the power law with cutoff.

The distribution of the pairwise CD for Nagoya is different than the one of the ICT. Even though the power law with exponential

1 Note that this p-value estimates the fitness of the model. In this case higher values indicate a better match [CSN07].

cutoff is still the best model matching the data (with $p - value = 0.99$ for all but one taxi) and the log-normal is the second model with identical p-values as with the ICTs, this time the Pareto distribution cannot be ruled out for a significant part of the taxis and provides a good match for all but 50 taxis (with $p - values$ varying from 0.2 to 0.85). Additionally, the log-likelihood ratio statistic indicates that the Pareto distribution fits equally well the empirical data for 113 of the taxis as the power law with cutoff. The parameters of the power law with cutoffs are exponent=2.42096 ± 1.049316 and rate=0.016773 ± 0.018044. The parameters of the log-normal distributions are mean-log=0.4774220 ± 0.2575515 and sd-log=$0.894645 \pm 0.22301747939546$. The parameter values of the aggregated CD for the power law with cutoff are exponent=2.260484 and rate=0.004525954, and for the log-normal - mean-log=0.4951964 and sd-log=0.9748073. The log-likelihood ratio points the power law with cutoff as the better fitting model.

6.3.2 *San Francisco*

The aggregated ICT and CD are shown in Fig. 63 and 64. Example pairwise distributions are given in Fig. 65 and 66.

The best matching model for the pairwise ICT distribution is the power law with exponential cutoff, which gives a $p - value$ of 0.99 on average for all taxis. The log-normal distribution comes second, by matching all taxis with a $p - value$ of 0.82 ± 0.1. The exponential and Pareto distributions are very poor fits of the data with no taxi been fitted with $p - value$ above the 0.01 significance level. The log-likelihood ratio confirms these findings by estimating the power law with exponential cutoff as the best fit for all taxis except 11, which are equally well fitted with the log-normal distribution. Until now the situation is equivalent to the Nagoya dataset. The parameters of the power law with exponential cutoff, however, are very different - exponent=0.289316 ± 0.1364428 and rate=0.0005083 ± 0.0005445. For the log-normal distribution mean-log=6.476539 ± 0.24252 and sd-log=1.586879 ± 0.091616. As with the Nagoya dataset, the log-normal parameters are in much smaller interval. The aggregate ICT distribution's estimated power law with cutoff parameters are exponent=0.3214054 and rate=0.0004494881, while the parameters for the log-normal distribution are mean-log=6.478439 and sd-log=1.578992.

The pairwise CD distribution also follows the same pattern as the Nagoya dataset. The power law with exponential distribution is the best fit for all taxis with $p - value > 0.99$, while the log-normal distribution is the second best-fitting with $p - value$ around 0.8 for all but one taxi, for which is 0.612, e.g. also a very good fit. As with Nagoya, the exponential distribution does not match the data at all, while the Pareto matches almost all taxis with varying levels of $p - value$ - from 0.01 to 0.791, with all but 15 taxis over the threshold of 0.2. As with Nagoya, the log-likelihood ratio estimated the power law with exponential cutoff as the best fitting model. Power law with cutoff's parameters are exponent=2.3623 ± 0.632605 and rate=0.018477 ± 0.02271, and log-normal's are mean-log=0.5627 ± 0.2218 and sd-log=0.998946 ± 0.174687. The distribution parameters of the fitted CD is much closer to the values of Nagoya's dataset, which implies that the contacts between taxis arise from similar traits of taxis' behaviors (contrary to ICT dis-

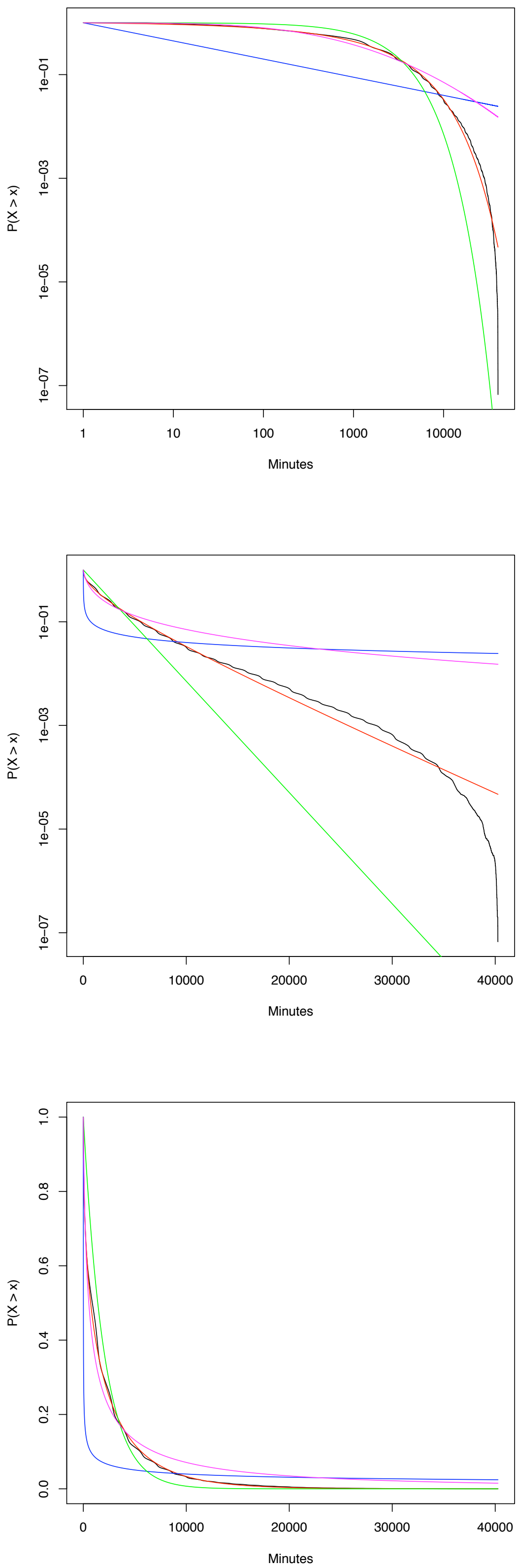


Figure 59: Nagoya dataset: Aggregated ICT over all taxis. The figures are in log-log, lin-log and lin-lin scales respectively. The black line represents the data, blue - Pareto, red - power law with exponential cutoff, green - exponential and magenta the log-normal distribution. The power law with cutoff fits best the data, followed by the log-normal.

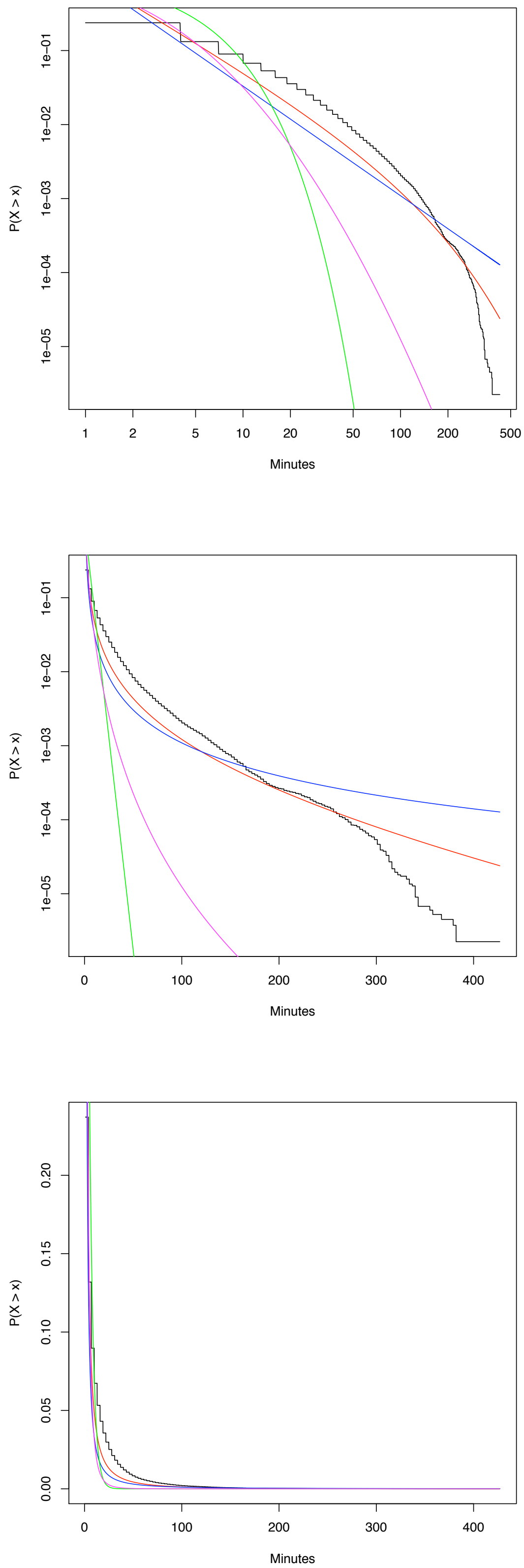


Figure 60: Nagoya dataset: Aggregated CD over all taxis. The figures are in log-log, lin-log and lin-lin scales respectively. The black line represents the data, blue - Pareto, red - power law with exponential cutoff, green - exponential and magenta the log-normal distribution. The power law with cutoff fits best the data, followed by the log-normal.

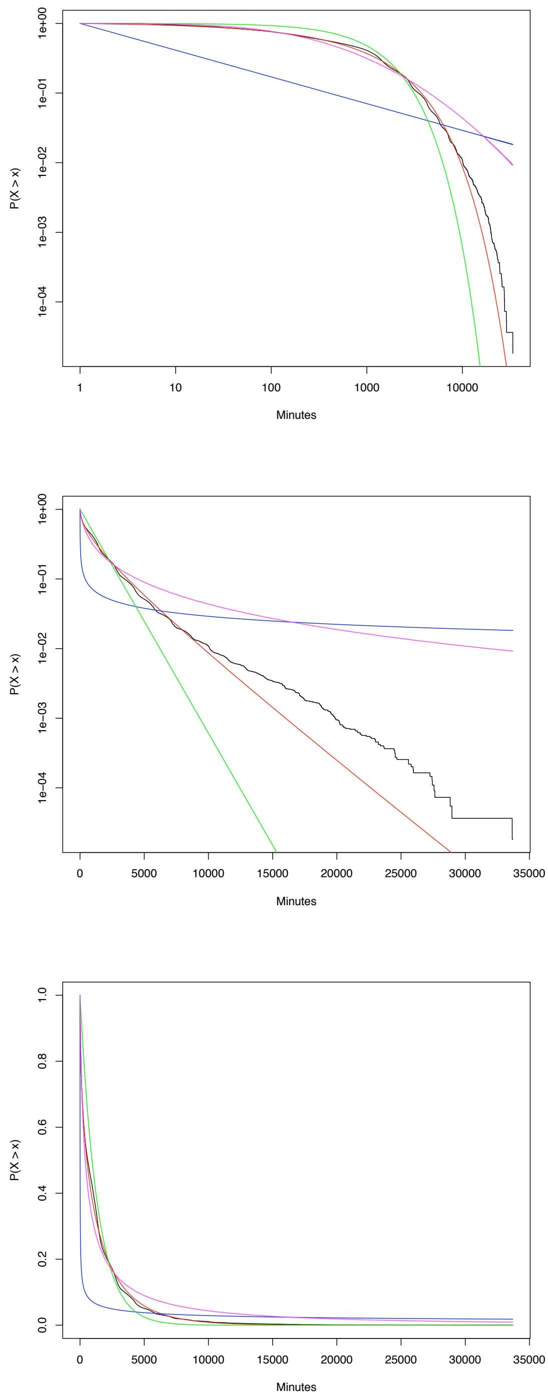


Figure 61: Nagoya dataset: Pairwise ICT for an example taxi. The figures are in log-log, lin-log and lin-lin scales respectively. The black line represents the data, blue - Pareto, red - power law with exponential cutoff, green - exponential and magenta the log-normal distribution. Here, the power law with cutoff fits best the data.

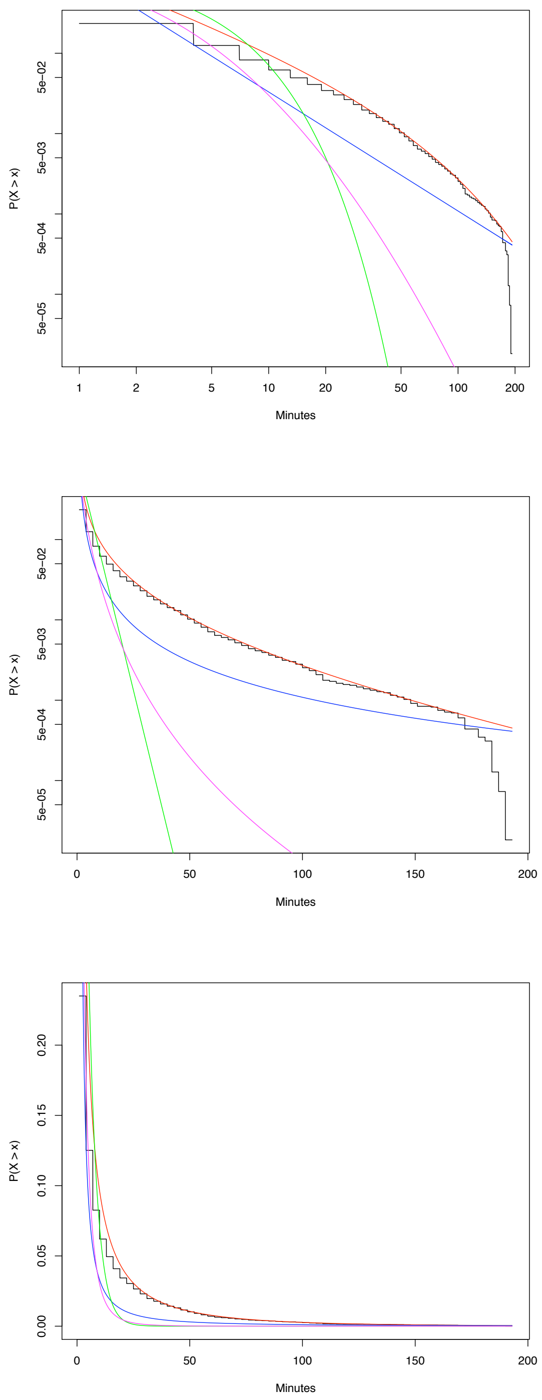


Figure 62: Nagoya dataset: Pairwise CD for an example taxi. The figures are in log-log, lin-log and lin-lin scales respectively. The black line represents the data, blue - Pareto, green - exponential and magenta the log-normal distribution. Here, the power law with cutoff distribution fits best the data.

tributions which differ significantly). The aggregate CD's power law with cutoff fitted model is with parameters exponent=2.530127 and rate=0.001205246, and log-normal's parameters - mean-log=0.5733612 and sd-log=1.045300.

6.3.3 *Context of validity*

Having analyzed the different ICT and CD distributions, we can define the contexts of validity for general taxis, taxis in Nagoya and taxis in San Francisco.

For a model to be valid in the context of general taxi, the pairwise ICT distribution must be a power law with exponential cutoff, with a good fit of log-normal distribution, and the pairwise CD distributions should follow power law with exponential cutoff, with good fit of log-normal distribution, and only occasional fit of pareto distribution. Additionally, the values of the power law with exponential cutoff fitting the CD distributions must be in the interval exponent=2.42096 $\pm$ 1.049316 and rate=0.018477 $\pm$ 0.02271, and the log-normal distributions with parameters mean-log=0.4774220 $\pm$ 0.2575515 and sd-log=0.894645 $\pm$ 0.22301747939546.

The Nagoya taxi context defines the same metrics to be evaluated, only with the parameters fitted to the Nagoya dataset. The same procedure is applied for the context of validity of San Francisco's taxis.

Additional characteristics can also be included in the context of validity, depending on the type of model being analyzed. For example, for a non-stationary model it would be a requisite to exhibit periodical behavior.

6.4 CONCLUSION

In this chapter we have analyzed two datasets of GPS traces gathered at Nagoya, Aichi, Japan and San Francisco, California, USA. The two data traces exhibit different movement patterns (such as number of trajectories), but in the same time turned out to follow common ICT and CD distributions.

We have described the data treatment procedure which allowed the proper analysis of the different categories. We have divided the data traces into runs, depending on the flag indicating whether the car is occupied or free, after which we have grouped the taxis based on the pairwise frequencies of encounters. Further, we have divided the time based on several factors, including heuristic and supervised clustering. This division of the data can be user to improve the accuracy of any model based on it.

Finally, the taxis of both cities exhibit diurnal patterns, but there are differences in the their characteristics. This is the first study that examines such a significant volume of unrelated GPS traces with the aim of explicitly defining a mobility model context of validation.

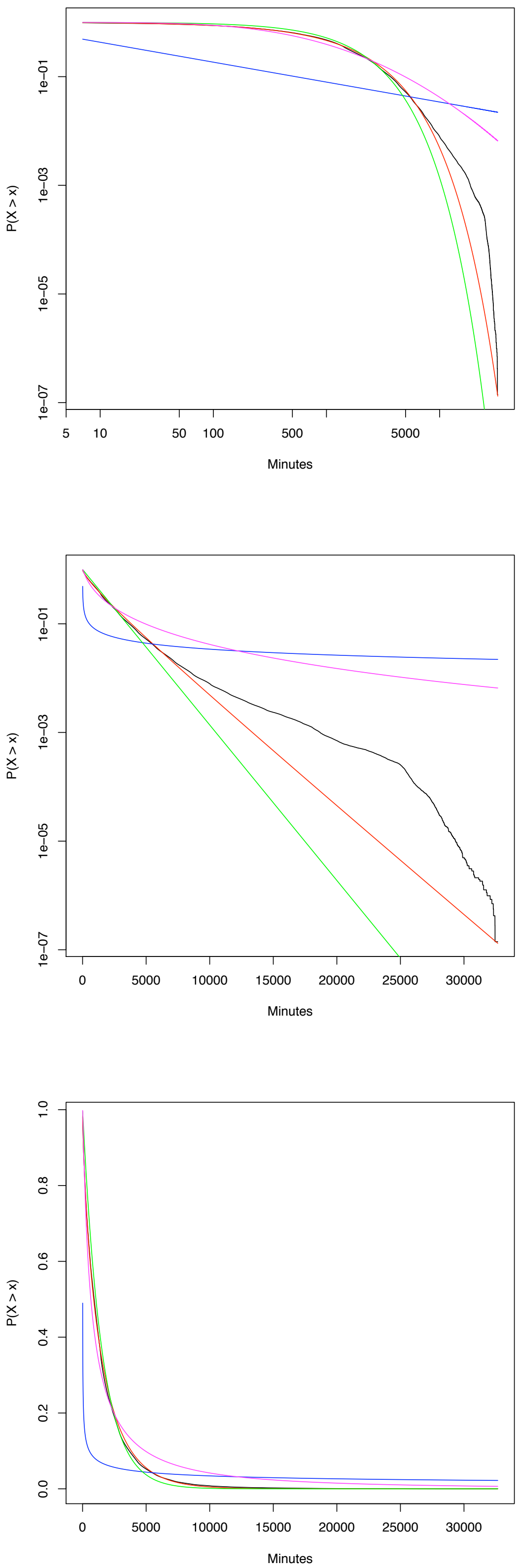


Figure 63: San Francisco dataset: Aggregated ICT over all taxis. The figures are in log-log, lin-log and lin-lin scales respectively. The black line represents the data, blue - pareto, red - power law with exponential cutoff, green - exponential and magenta the log-normal distribution. The power law with cutoff fits best the data, followed by the log-normal.

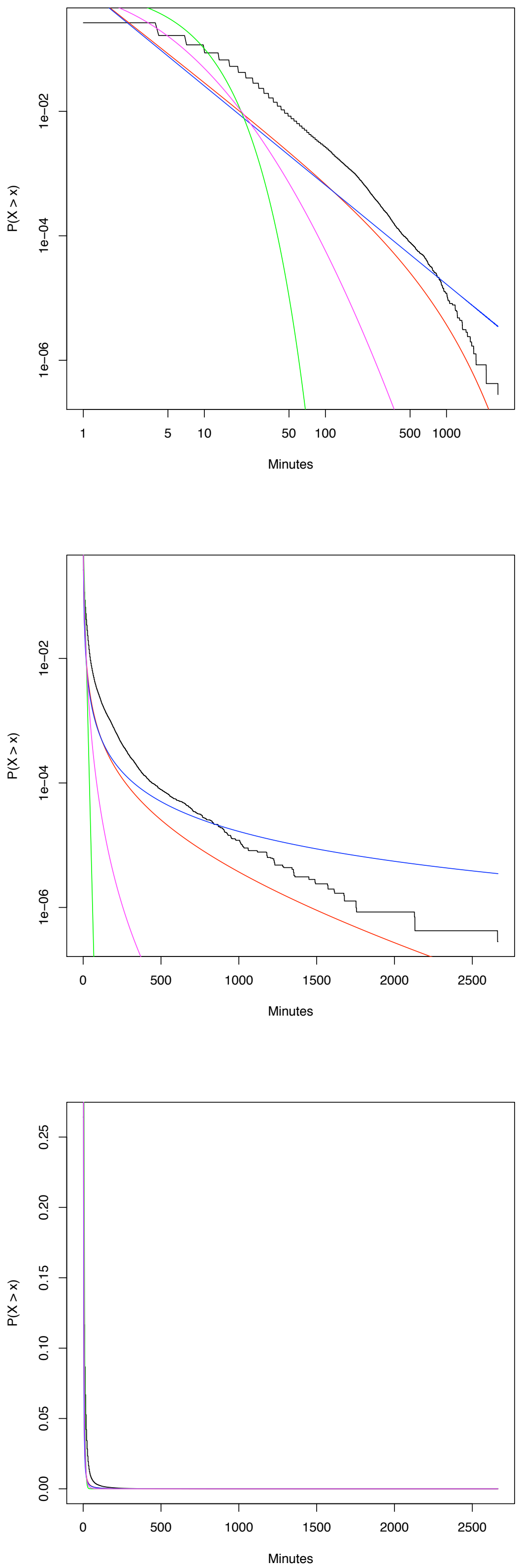


Figure 64: San Francisco dataset: Aggregated CD over all taxis. The figures are in log-log, lin-log and lin-lin scales respectively. The black line represents the data, blue - pareto, red - power law with exponential cutoff, green - exponential and magenta the log-normal distribution. The power law with cutoff fits best the data, followed by the log-normal.

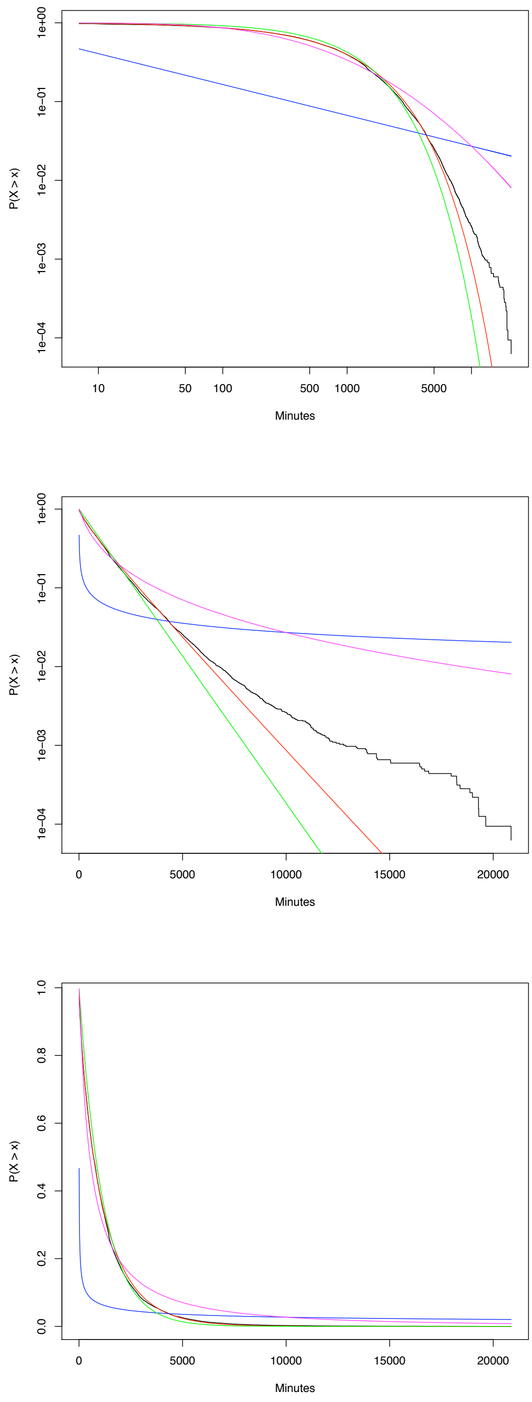


Figure 65: San Francisco dataset: Pairwise ICT for an example taxi. The figures are in log-log, lin-log and lin-lin scales respectively. The black line represents the data, blue - pareto, green - exponential and magenta the log-normal distribution.

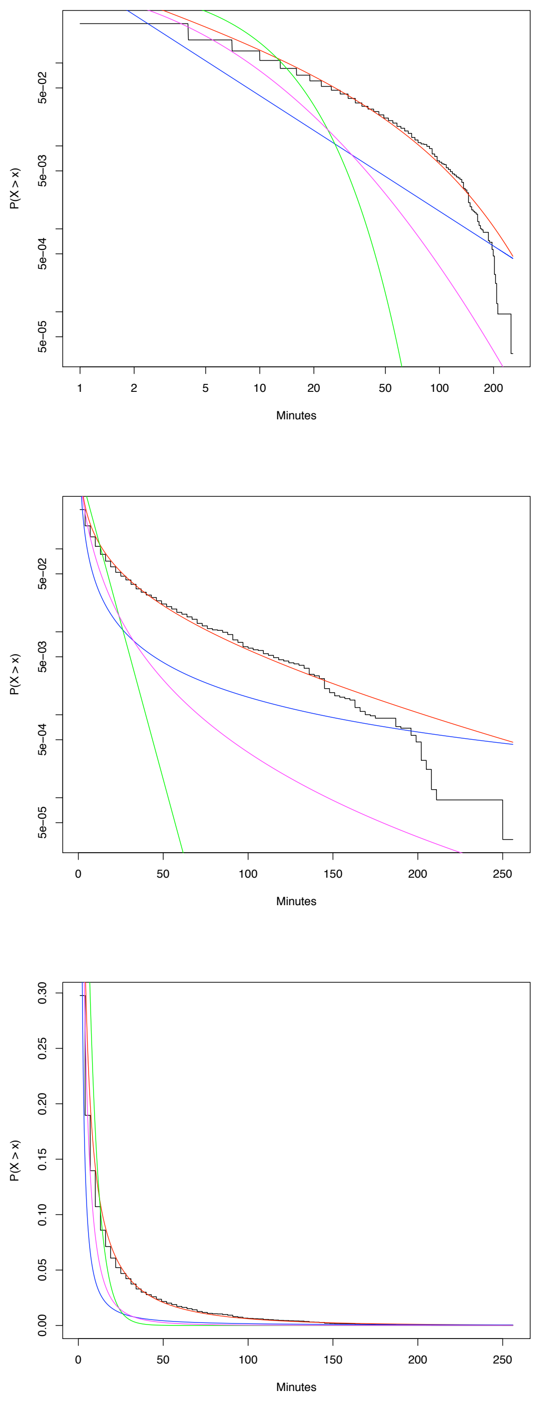


Figure 66: San Francisco dataset: Pairwise CD for an example taxi. The figures are in log-log, lin-log and lin-lin scales respectively. The black line represents the data, blue - pareto, green - exponential and magenta the log-normal distribution.

Part IV

CONCLUSION

7 GENERAL CONCLUSION

7.1 CONCLUSION

Mobility is an important characteristic of wireless networks, and as such has to be correctly modeled and well understood. In this thesis we focused our efforts into defining what should be the goal of such mobility models, how can we judge if a model faithfully represents a given real-world situation, how can we easily build such models, and how can the mathematical properties of these models be studied.

Recognizing that mobility models are mostly used as tools by wireless network researchers is an important step towards understanding which are the relevant properties that need to be improved. In the introduction chapter (Chapter 1) we summarized that such model should ideally be easy to use, understand and modify. Additionally it should have understandable parameters and known mathematical properties, and be validated against some real-world traces.

Chapter 2 introduces the different types of model validation and verification procedures and the way the can be interpreted in the context of mobility modeling. Verification methods are aimed at assuring that the model has been implemented correctly and that it performs as expected. Validation on the other hand treats the procedures related to the model definition. The chapter is based on the simplified version of general modeling process and addresses conceptual, data and operation validity. *Conceptual validity* assures that the correct assumptions and theories have been applied, and that all related prerequisites are met. The correctness and the adequacy of the data used for the model definition and validation concerns the *data validity*. Finally, we have formalized the general mobility model validity based on the *operational validity*, which uses model's output to determine if its behavior is as expected. Thus, instead of relaying on an informal concept of model realism, a model should be validated (by using appropriate metrics) in a given context (acceptable metrics values).

Having defined the approaches to model validation, we have surveyed the existing mobility modeling approaches in Chapter 3. We aimed our study on the models themselves, and other related data-based works that are relative to real-world mobility patterns, presented in Section 3.3. The models themselves have been categorized into synthetic (Section 3.1) and empirical (Section 3.2), with some additional models presented in the appendix (Sections A.2 and A.3). Synthetic models are based on simple rules and/or mathematical principles, and are typically aimed at representing idealized mobility patterns with none or limited context of validity. They are further subdivided in analytical and physically-based models depending on the characteristics upon which they have been defined. Alternatively, the empirical models are founded on traits extracted from heavy collections of real-world data. The used data may be coarse-grained (e.g. WLAN) or fine-grained (e.g. GPS) traces, or a real-world map. Even though empirical models are often accepted as realistic (in the informal sense) they are rarely used with the RWP being one of the most popular models to be used [KCC05].

This is partly based on the complexities related to using models which are not readily available, have involved descriptions and unknown mathematical properties, and are difficult to adapt to other use cases.

Chapter 4 is dedicated to the introduction of our proposition - the Layered Mobility Model Architecture (LEMMA). It is founded on few simple core principles, which after analysis have been shown to provide a wealth of possibilities to model users and creators. Under LEMMA, the node environment is standardized with the help of four types of components - the universe where the movements occur (the *simulation area*) is divided into different regions (*zones*) which determine the coarse-grained node movement patterns. The movements are restricted by the movement *constraints* and are further refined with the help of additional *influencing factors*. Following the standardization of the environment, we have divided the movement process into five semantically distinct, strictly defined layers. The layer communicate via simple interfaces in a unidirectional (top-to-bottom) manner and deal with separate characteristics of the movement patterns. The high-level decision making process is called *strategy*. It operates at the level of *zones* e.g. it governs the coarse-grained behavior of the nodes. Afterwards, the *mapper* layer translates the high-level movements into low-level (coordinate-based) positions and passes them to the *tactic* layer, which selects the route to be taken towards the destination point. The *dynamics* layer adds the temporal dimension of the movement by specifying the speed at each point of the trajectory. Upon reaching the destination, the *stay* layer determines the motion of the node for a period of time fixed in advance by the *strategy*. By analyzing the consequences of these principles, we have shown that LEMMA clearly separates the scope of action of the different layers (4.4.1), incites movement principle generalization (4.4.2), and layer aggregation (4.4.7), and facilitates model verification (4.4.3) and validation (4.4.4). Additionally, we have shown that it provides a universal framework for mobility model creation (4.4.6) and is indeed realistic and feasible (4.4.10). We have further discussed the possibility of creating advanced models which exhibit group (4.4.9) and heterogeneous (4.4.8) movement patterns. These characteristics respond to the problems hindering the introduction models which are more advanced, validated, and are easy to use and to modify and understand.

The mathematical foundations of LEMMA have been the object of study in Chapter 5 where we have shown the way deterministic and probabilistic models can be defined by following architecture's principles. The chapter provides the necessary base for defining mathematically tractable models with LEMMA. Additionally, we have proved several important theorems providing the formal evidences that any mobility model can be represented in a non-trivial way by LEMMA, and that there are no internal inconsistencies in the architecture (i.e. all combinations of valid layers induce a valid mobility model). Furthermore, we have provided an example of important stochastic property which can be derived for the whole model by only knowing the properties of its constituent layers. Indeed, knowing that a model possesses a stationary regime by simply proving stationarity of its layers is a unique feature of LEMMA and extends further the possibilities provided by layer recombination.

Finally, in Chapter 6 we analyzed two different scenarios - taxi movements in Nagoya, Aichi, Japan and San Francisco, Callifornia, USA. We have determined the context of validity for the two environments based on several wide-spread metrics (distribution of the aggregated and pairwise ICT and CD).

In conclusion, we have structured this thesis by analyzing the needs of model creators and users and by trying to find a solution addressing both worlds without sacrificing its generality or accessibility. We have studied what are the traits making a mobility model valid. We have conceived LEMMA, which is a universal framework built on few simple principles, and in the same time providing a multitude of benefits. Apart from being a flexible framework upon which may be built any mobility model, be it individual or group, synthetic or empirical, or homogeneous or heterogeneous, the architecture provides the means to mathematically analyze it and in some circumstances may indicate its stochastic properties. In order to demonstrate the applicability and the feasibility of the architecture, we have developed an open-source simulator.

7.2 FUTURE WORK

Mobility has become and is going to remain a inseparable part of wireless network research. No more is the simple RWP sufficient to satisfy the criteria of acceptable mobility scenarios - it has to be augmented with other mobility patterns which can show additional, potentially more plausible, weaknesses or strengths of the evaluated solutions. For that reason extending and improving the LEMMA simulator can be of great benefit to the entire research community, as it is backed by a stable theoretical framework.

The mathematical properties of the architecture need to be further studied, as there is a lot of potential in discovering model stochastic properties based on layers' characteristics. We are studying model ergodicity and closed-form representation of the stationary distribution, but these may be extended to any other aspect of models' nature.

The possibility to create per-layer validation and verification procedures open a wide horizon of possible research opportunities - defining such procedures and studying the relation between per-layer validation and model validation has the potential of bringing many interesting results, both from theoretical and from practical point of view. The same principles may be used to directly compare different layer implementations of the same layer, or provide approximation procedures and estimations given some global model restrictions, e.g. find a markov chain estimation of minimal order of a given strategy, such that given the same lower layers, certain metrics would remain stable.

Finally, we have only made the first step in the analysis of the two datasets presented here. Including additional characteristics to the analysis may change the model in question. Furthermore, a study of competing models matching the data may find a simpler model, or a model which more accurately reflects some of the movement patterns. Currently we are working on the definition and validation of several new mobility layers which are valid in the context of the two taxi traces we analyzed in Chapter 6.

Part V

APPENDIX

APPENDIX A

A.1 EXISTING LAYERS

This section presents an outline of the various environment components and model layers that have been used for the definition and implementation of different existing mobility models. The properties and the typical use cases of each layer have been discussed.

A.1.1 *Environment Components*

Most of the wireless network simulations are performed in a 2D simulation area, which is to some extent determined by the limitations of one of the most popular network simulators (*ns-2*), but also because of the complexities resulting from the usage of a 3D simulation area. However there are some exceptions, such as the swiss flag and the fish bowl [LBV06a], and the 3D parallelepiped [TG05]. The typical border behaviors include bounce-back or wrap-around.

Although the definition of a zone allows for almost any object to be represented in great detail (for example following the shape of the walls in a house), the only kinds of objects used for simulations are the simple geometric shapes - rectangles, cubes, circles and singletons, with the notable exception of [Sio05] where any connected set of points may be used. Indeed, that kind of details is rarely necessary given that the mobility models themselves include a lot of simplifications and approximations.

The number of constraint types may seem potentially very large, but in the majority of the cases is limited to three kinds - zone avoidance, zone confinement and graph confinement. Each of these constraints limits the movement of the nodes in the obvious way - zone avoidance restrains the conforming nodes from entering into zones marked to be avoided, zone confinement keeps the nodes in their allowed areas, and graph confinement restricts their movements on the edges of an embedded in the simulation area graph. What most of the papers do is define ways of parametrizing these constraints (e.g. building graphs by using Voronoi paths [JBRAS03], Delaunay triangulation [Hua05], synthetic maps [BSH03], real-world maps [NKK02, SJ04, NBG06, YNLK06, Bra99]).

Movement influencing factors are used primarily for scenarios aiming at increased model realism. These are almost always aimed at VANET simulations and include traffic signalization [Mar97, KB05, IR05, PM06] or regulating the speed as a function of the surrounding nodes [TZH+02, SJ04, Ste02, THH00, JWZ01, LWM03, TKH06].

A.1.2 *Strategy Layers*

A.1.2.1 *Manually Specified*

Parameters: Zones, stay times.
Main idea: Visit the predefined zones in order.

The manually specified strategy takes two parameters - the list of zones to visit and the list of stay times, both of which are defined with a strict order, which is respected during the execution of the layer. The algorithm itself is straightforward execution of the given scenario, that is - the zones and stay times are returned in the order they appear in the parameters. Once the given zones are visited the execution of the layer restarts from the beginning of the list. The movement of the logical centers of the groups of the model presented in [HGPC99] is an example of this strategy.

This strategy may be used when the exact behavior of the nodes is known, or at least when their high-level movements have to be fixed for a deterministic scenario. It may be also used for testing specific rarely-occuring cases, which would require multiple simulations in order to appear in a purely probabilistic simulation. Also, replaying real-world traces could also be achieved with the help of this layer, e.g. replaying dense (not sporadic) Wi-Fi association traces or cellular network mobility traces. Another possibility is to simulate group mobility by setting the same fixed set of parameters to several nodes. This however is not the native way of achieving this in LEMMA, as layers can be shared across node mobility models.

The strengths of this layer may be regarded as weaknesses in some cases - the scenarios defined with it are going to always exhibit the same high-level behavior. Experimenting with several closely related situations would require redefining the scenario, which could be a tedious and error-prone process. Also, blindly replaying real-world traces without understanding of the processes that generated these traces may lead to biased results.

A.1.2.2 *Uniform Random Zone*

Parameters: Zones, minimal and maximal stay times $(P_{min}; P_{max})$.
Main idea: Select the next zone and stay time randomly with uniform distribution.

The parameters to this layer are the list of zones accessible for the node and the possible values for the stay time (given as an interval). The next zone to be visited is selected from the list of zones with uniform probability. The stay time is drawn with uniform distribution from the interval $(P_{min}; P_{max})$. Examples of mobility models which can be constructed with this layer include the Random Waypoint Mobility Model (RWP) [JM96] and the Restricted Random Waypoint Model (RRWP) [BGB01].

This is a simple strategy that may be used in basic, synthetic scenarios where the realism of the high-level movements is not the most important aspect of the simulation. It may also be considered as the first step towards a more complex node behavior, or even a base for comparison (for example the majority of the MANET routing protocols were evaluated with such simplistic models, so comparing against their performance would require either repeating all simulations with the new mobility models, or copying the setup in which they were executed).

The next zone to be visited is chosen from the list at random, and the stay time is selected in the range $[P_{min}, P_{max}]$, both with a uniform distribution. This is the strategy corresponding to the omnipresent Random Waypoint Mobility Model (RWP) [JM96].

A.1.2.3 *User-distributed Random Zone*

Parameters: Zones, next zone distribution Z_D, stay times distribution P_D.
Main idea: Select the next zone and stay time randomly from user defined distributions.

As any other strategy, this layer also takes the list of zones from which the next zone is to be chosen. It uses node's current zone (z) and the current simulation time (t) in order to determine the distribution Z_D from which to select the next zone (thus $Z_D = Z_D(z, t)$). Afterwards, the stay time is determined based on the destination zone d selected in the previous step. The stay time is randomly selected with distribution $P_D(d)$. This strategy is used for the construction of the Weighted Waypoint Mobility Model [HMS+04,HMS+05].

This layer can be used to create time-varying behavior (by specifying different distributions for the different intervals of the simulation). However this type of behavior may be accomplished by using LEMMA's possibility to create hybrid mobility models and replace different layers upon the occurrence of certain events which include time change. Furthermore, this layer may be used to create uneven node spatial distribution, e.g. by defining "hot" zones to be visited with greater probability, and to contain the nodes for longer periods. It should be noted that this layer can be reduced to a first order Markov chain.

The difficulty of this layer lie in the specification of the distributions, which increases with the number of zones in the scenario and the number of different time periods that have to be simulated. This strategy may be considered a generalization of the *Uniform Random Zone* strategy, which may be obtained by setting both Z_D and P_D to uniform, time-invariant distributions. It provides more freedom to the scenario creator and is more appropriate for modeling empirical observations or theoretical models.

A.1.2.4 *Uniform Random Direction*

Parameters: Zones, vector function returning all zones in a given direction, minimal and maximal stay times.
Main idea: The destination zone is selected from all zones that are in a randomly selected direction.

The algorithm starts by generating a random vector d. The vector function F given as parameter returns all zones in the direction determined by the random vector d. The destination zone is then selected with uniform probability from that list. The stay time is selected uniformly in $[P_{min}, P_{max}]$. This strategy corresponds to the Random Walk Mobility Model (RW) [BNKS95].

The Random Walk is a very important case of the mobility models. It has been extensively studied from different perspectives, including physics and economics. It is a simple purely synthetic model that is rarely going to correspond to the movement of something else than a molecule. However, it is no worse than the RWP (e.g. Uniform Random Zone strategy) and in many cases has more desirable properties (being in a stationary mode under most circumstances [BV04]).

A.1.2.5 *Mathematical Function*

Parameters: Zones, vector function.
Main idea: Evaluate the function and select a zone containing its value.

The mathematical function layer takes a vector function $F(t)$ returning a point of the simulation area for each time instant t. Afterwards, the next zone is the one with the smallest volume containing the given point. If no zone contains the point for the exact moment t, the function is evaluated for values $\tau_i > t$ until the function gives a point in a zone. Examples of the usage of this strategy include the Random Gauss Mobility Model [Tol99] or the Exponential Correlated Random Mobility Model [B+96].

This strategy allows the usage of analytical models in the high-level decisions of a simulated mobility model. Using equations modeling particle flows leads to the classical case of producing microscopic simulations by sampling the output of the macroscopic solution. Nevertheless, the models using this layer are either overly simplistic (e.g. linear equations), or with multiple, hard to configure parameters (e.g. a single parameter changing the movement of the nodes, but without a direct physical meaning).

A.1.2.6 *Activity-based*

Here we are going to describe the approach as defined by Scourias et al. [Sco97, SK99a, SK99b].

Parameters: Zones, typical zones, list of activities, activity transition and duration matrices.

Main idea: The next activity is selected, then its duration. The next zone to be visited is then chosen depending on that activity.

The algorithm used by the layer first selects the next activity to be performed. It is selected randomly from the activity transition matrix, given the current activity and time of day. Its duration (i.e. the *stay time*) is chosen from the activity duration matrix. There are three cases which are used when selecting the *destination zone* :

- *If the node has a typical zone for this activity* - select it as the destination zone.
- *Else, if there are zones having weights associated to this activity* - order them in descending order according to their weights divided by the distance to them, and randomly pick the destination zone from the top five.
- *Otherwise* - randomly select the destination zone.

The activity-modelling strategies are all aimed at providing realistic high-level node behavior. The realism depends on the data available to the researcher - there is an abundance of surveys performed in different areas of the world. However, the results of these surveys are not always readily available, which makes the usage of this strategy difficult in these cases. Also, as with all historic data-driven models, activity-based models only reflect the past and current behavior of the people. Thus, trying to simulate their future behavior, or the actions in an extreme situation (as for example during an earthquake rescue operations) may require data extrapolation or heuristic guesses of the non-negligible number of parameters.

A.1.2.7 *WLAN Trace-based*

Here we are going to describe the approach as defined by as defined by Tuduce et al. [TG05].

Parameters: Zones, stay time distribution, probabilities to stay in the current zone, or move to a neighboring or non-neiboring zone.
Main idea: The next zone if selected based on the probabilities.

The principle of the strategy is to select the next zone to be the same as the current zone (with probability p_{same}), to be a neighboring zone (p_{neigh}) or a non-neighboring zone ($p_{non-neigh}$). Then, the stay time is drawn from fixed distribution.

This strategy was introduced as a model for user Wi-Fi associations. It is one of the few data-based models that have such concise description, and so little and straightforward parameters. The type of movement that is captured reflects the fact that people are more likely to stay at the same place, or move to a neighboring one, than to go to a more distant location. However, the notion of typical sojourn places is missing (for example office/home) and it might be impossible to reflect such kind of complex behavior. This strategy models very well in-building individual movements, but it may prove to be less adapted when considering bigger scale movements with patterns.

A.1.3 *Mapper Layers*

A.1.3.1 *Fixed*

Main idea: The point to be selected is drawn from a list, which is specified by the creator of the scenario.

This mapper resembles the manually specified strategy, only applied to a lower layer. However, there are several possible ways to define the points to be visited - for example one may provide the list of fixed points to be visited, along with a probability associated to each of these points. Each of the points is then selected with its corresponding probability. Another possibility is to specify the exact order of the points to be selected.

The typical usage includes the modeling of passageways (doors, checkpoints, tunnel entrances or exits, ...) or typical dwelling places (office desks, chairs, sofas, etc.) as for example in [MMLR05].

A.1.3.2 *Uniform Random*

Main idea: Randomly select a point in the given zone with a uniform distribution.

The Uniform Random mapper is frequently used as for the non-stationari RRWP [BGB01] and for the non-stationary RWP [JM96]. The destination point is chosen randomly in the destination zone, which is easy to implement for any shape of the zone. It does not allow the formation of clusters of the arriving nodes, as the assigned destination points are uniformly distributed.

A.1.3.3 *User-distributed Random*

Main idea: Randomly select a point in the given zone with a user-defined distribution.

This layer is a generalization of the Uniform Random mapper. With it, the scenario creator may specify the probability distribution for the points for each of the zones. Each zone is being assigned a distribution, which gives great level of flexibility. This allows the simulation of cities with areas of concentration (malls, residential areas), which cannot be reduced to single points (and consequently cannot be modeled with the Fixed mapper). This mapper may be used in order to obtain the stationary regimes of the RWP and the RRWP [BV04].

A.1.3.4 *Random Border*

Main idea: Randomly select a point belonging to the border of the given zone.

This is the mapper used in the Random Direction Mobility Model ([RMSM01]) or in [HR86]. It may be used in any scenarios where the movements of the node are different in a given zone when compared to the exterior - as for example the area around a disaster, or a residential area.

A.1.3.5 *Gravity Center*

Main idea: Select the point of gravity of the given zone.

The center of gravity of a zone 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. The center of the gravity for a triangular zone is its centroid. For a rectangular zone that is the intersection of the two diagonals. An example of its usage may be found in [SK99a]. This mapper is a specific case of the Fixed mapper, but is much easier to use and configure.

A.1.4 *Tactic Layers*

A.1.4.1 *Linear*

Main idea: Movement in a straight line.

The node moves in a linear manner (a straight line) from its current location to the destination. Used in a wide variety of models, such as [HR86, BCSW98, JLH+99b, BGB01, TG05]. This tactic does not enforce any constraints and is suitable for simple scenarios where no obstacles or other form of movement restrictions are required. However, trace based or any other real-world based scenarios are less likely to be accurately represented by this type of route selection.

A.1.4.2 *Zone Avoiding - Shortest Route*

Main idea: Take shortest routes avoiding some of the zones.

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

This tactic enforces *zone avoidance* constraints, where one or more of the zones are marked to be avoided with an attribute. As an example one may consider modeling a disaster area by marking some of the zones as “ruins" and then assign this tactic to the rescue vehicles, which would forbid them entering these zones.

A.1.4.3 *Zone Avoiding - Border Route*

Main idea: Move linearly towards the destination but avoiding some of the zones by moving on their borders.

This layer chooses linear routes, but it avoids the zones marked as unaccessible. Instead, the node moves in straight line directly towards its endpoint until it reaches an unaccessible zone, which is then surrounded 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.

This algorithm, like A.1.4.2, enforces the *zone avoidance* constraints with the difference that nodes do not take the shortest routes, but instead surround tightly the zones. The choice between this tactic and A.1.4.2 depends on the specific scenario. For example, when compared to the same example, this tactic would make all rescue vehicles pass by all marked zones in the way.

A.1.4.4 *Zone Confining - Shortest and Border Route*

Main idea: Confine the movement to certain zones by taking shortest or border routes.

Analogically to the *Zone Avoiding* cases, only that nodes are allowed to move only on the zones marked as accessible. The tactic enforces the *zone confinement* constraints. Considering the same example as in the previous two tactics, there may be rescue teams with dogs, or medics, that can move only with the zones having the attribute "ruins", e.g. [AFMT04].

A.1.4.5 *Graph-constrained - Shortest Route*

Main idea: Follow the shortest route of a map, represented as a graph.

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.

This algorithm enforces the *embedded graph* constraints. 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). This route selection is the predominantly used in VANET simulations. Once it has been configured it provides much more realistic route selection than the simple linear movements. It can also be used to simulate obstacles. One of the major difficulties is specifying the graph itself. There have been several works aiming at synthetic route generation [HMZ05], using graphs following some patterns (like the City Section Mobility Model [Dav00, CBD02], or using real-world maps such as the ones given by the U.S. Census Bureau [Cen] or the Japan Digital Roadmap Association [drm]. In general, using this tactic increases the complexity of the scenario, but can potentially increase its realism. Using it in the best way requires careful planning and understanding the modeled environment.

This layer can be further generalized, as for example in the *Graph-constrained - Shortest* N *Routes* tactic (used in [YNLK06]), where instead of always selecting the shortest route in the graph, one of the top N (given as parameter) shortest routes is randomly chosen.

A.1.4.6 *Graph-constrained - Location Visiting*

Main idea: Follow the routes of a map, represented as a graph, but also visit some intermediate locations.

This algorithm requires an additional parameter, when compared to A.1.4.5 - the list of intermediate locations to be visited. These locations vary depending on the start and end vertices. The locations are visited in order by using the *graph-constrained - shortest route* tactic.

Used in [Sco97,SK99a,SK99b], this tactic allows adding some complexity to the route selection routine. For example it allows the specification of a route preference (when going home from work skip the freeway and go through the center), or adding asymmetric route selection (when going work from home always take the freeway). It has been shown that people may take different routes when doing a round trip [Gol95]. One of the difficulties with this tactic is that in addition to having to specify the constraining graph, one has to define the list of intermediate points, which increases the complexity of configuring the simulation. Furthermore, selecting a different graph would require a redefinition of the list.

A.1.5 *Dynamic Layers*

A.1.5.1 *Constant*

Main idea: Move at a constant speed.
Parameters: Minimal and maximal velocity.

This dynamic first draws a velocity uniformly from the interval $[V_{min}, V_{max}]$. Then, the node moves constantly on the trajectory until reaching the destination point. A new speed is drawn for each trajectory.

One of the most frequently used dynamics [HR86, BCSW98, JLH+99b, TG05, HMS+05] mainly because if its simplicity. Furthermore, this dynamic is appropriate when the high-level movement patterns are more important than the low-level movement interactions. It does not provide any internode interactions nor any variations in the speed, which in some cases may compromise the realism of a model whose upper layers are deemed “realistic" (e.g. a congestion can never occur, even with all vehicles passing through a single crossroad in the same time). Another possible pitfall is the speed decay that may occur with this dynamic. This is the phenomenon where nodes get trapped into long trajectories moving at slow speed, which in turn decreases the average node speed.

A.1.5.2 *Acceleration-constant-deceleration*

Main idea: Accelerate, move at a constant speed, and finally decelerate.
Parameters: Minimal and maximal velocity.

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

This is a modification of the *constant dynamic*, which aims at providing a gradual change in the speed instead of instant speed jumps. However it does not resolve any of the other issues. In certain circumstances it may be considered a good approximation of the real-world, such as for airplane or train simulations.

A.1.5.3 *Random*

Main idea: Randomly change the current speed.
Parameters: Minimal and maximal velocity, and 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 $[V_{min}, V_{max}]$. An example of its usage may be found in [BSH03].

This dynamic alleviates the issue of speed decay existing with the *constant dynamic*, and has the obvious advantage of being simple. Furthermore, it is possible to write a closed form expression for its expected speed, acceleration, etc. By assigning different values to the acceleration and velocity limits one may try to simulate different types of nodes (pedestrians, vehicles, etc.), however the constant changes in the velocity are not characteristic for the majority of the entities of wireless network simulations.

A.1.5.4 *Smooth Random*

Main idea: Select a target speed at random intervals and accelerate until reaching it.
Parameters: Set of target 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 new target speed is drawn by a random process 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). This method was introduced by Bettstetter [Bet01b].

This layer is a smoothened version of the purely random layer introduced in A.1.5.3. As such, it alleviates the issue with the possible erratic behavior of the nodes. However, analyzing it mathematically might prove to be more difficult depending on the nature of the random process selecting the target velocity. Even though by choosing the correct set of target velocities may provide realistic behavior for some situations, it should be used with caution, as internode interactions are ignored.

A.1.5.5 *Edge-limited*

Main idea: Limit the speed when there are too many other nodes on the same edge of the graph.

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 or congestion with little complexity. Introduced by Breyer et al. [BKOKR04].

This dynamic uses the underlaying graph structure to simulate congestions/overcrowding and thus is more appropriate to be used when these events are expected to occur often. Choosing the edge-limited dynamic to act as an upper bound of another dynamic (as A.1.5.4 for example) may add taking internode interactions into consideration to it, but at the expense of the mathematical tractability.

A.1.5.6 *Intelligent driver*

Main idea: Each node constantly adjusts its speed depending on the distance to the node preceding it.
Parameters: Safe time headway (T), maximal acceleration (a), desired deceleration (b), acceleration exponent (δ), jam distances (s_0, s_1).

This dynamic layer implementation is specific to simulations of vehicles [THH00]. It belongs to the family of car-following models, which have been extensively studied by the traffic research community. The desired velocity v_0 may be fixed for the whole simulation, or vary depending on the speed limitations of the given road. Each vehicle constantly adjusts its acceleration according to a continuous function $\dot{v}$ of its velocity v, and the gap s and velocity difference Δv to the leading vehicle:

$$\dot{v} = a[1 - (\frac{v}{v_0})^{\delta} - (\frac{s_{dmg}}{s})^2],$$

where s_{dmg}, the *desired minimal gap*, is a function of v and Δv:

$$s_{dmg}(v, \Delta v) = s_0 + s_1\sqrt{\frac{v}{v_0}} + Tv + \frac{v\Delta v}{2\sqrt{ab}}.$$

This layer can be used when congestion and other traffic related characteristics have to be modeled accurately. However it is more difficult to use, as the parameters have to be selected appropriately depending on the simulated environment. Additionally, the usage of this dynamic would require that the position of all nodes is recalculated for each time instant of the simulation, which may considerably slow the execution of the simulation. Other examples of the same family of traffic-related dynamic layers include the Velocity-Difference Model [JWZ01], the Bounded Rational Driver Model [LWM03] and the Human Driver Model [TKH06].

A.1.5.7 *Oscillating*

Main idea: Oscillate around the positions given by the base dynamic.
Parameters: Base dynamic.

The base dynamic generates its movement instructions, from which is sampled a set of intermediate points P. The node visits the points from the modified set $\{p + d(p) | p \in P\}$, where $d(p)$ is a random vector. The velocity is adjusted to reflect the change in the length of the trajectory.

The oscillating layer can be used to add some randomness in the movement of a linear (but possibly realistic) tactic. Because the output of the base dynamic is sampled in multiple points, and scenario creators may limit the length of the random vector d, the extent of modifications can be limited to a desired precision (if d is set to be always 0, then the output of the base dynamic will be unchanged).

A.1.5.8 *Reference Point Group*

Main idea: Add the fixed vector to the generated movement from the base dynamic (which is normally shared by several nodes).
Parameters: Base dynamic, fixed vector.

The base dynamic D generates the base movement instructions to be followed by all nodes from the group. A set of intermediate points P to be visited by the node is sampled from these instructions. The node visits the points from the set $\{p + \rho + d(p) | p \in P\}$, where $d(p)$ is a

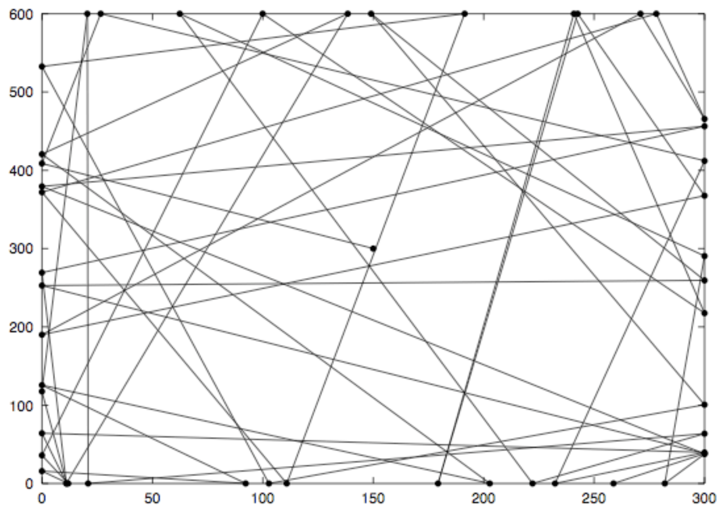


Figure 67: Trajectory of a node moving with the RD movement pattern in a 2D area (From [CBD02]).

random vector and ρ is the fixed vector, possibly different for each of the nodes. The velocity is adjusted, so that the node reaches the points for the time indicated by D.

This dynamic allows scenario creators to simulate node group movement. By assigning a movement process using this dynamic to several nodes and accordingly setting the radius vector of each of them, one ensures that they will move in a formation. By changing the way random vectors are generated one may obtain more or less compact groups, or change the degree of freedom of individual nodes. This layer can be used to construct the Reference Point Group Mobility Model [HGPC99].

A.2 ADDITIONAL SYNTHETIC MOBILITY MODELS

A.2.1 *Analytical Models*

A.2.1.1 *Random Direction (RD)*

The node selects a destination point on the border of the simulation area, after which it moves to it with a speed chosen uniformly in a given interval. Once it has reached the destination, it pauses for a random amount of time, also selected with uniform probability in a given interval (shown in Fig. 67). The model was used in [HR86] for cellular networks, where the movement patterns changed on the borders of the cells. A modification of the model includes changing the movement direction anywhere in the movement area, not only on the borders [Gue87, RMSM01].

A.2.1.1.1 SMOOTH

Introduced in [Bet01a, Bet01b], the model is an enhancement to the RD, making the node movement "smoother" (Fig. 68). It modified the way both speed and direction are changed. The speed and the direction are changed separately, as indicated by two random processes. The process

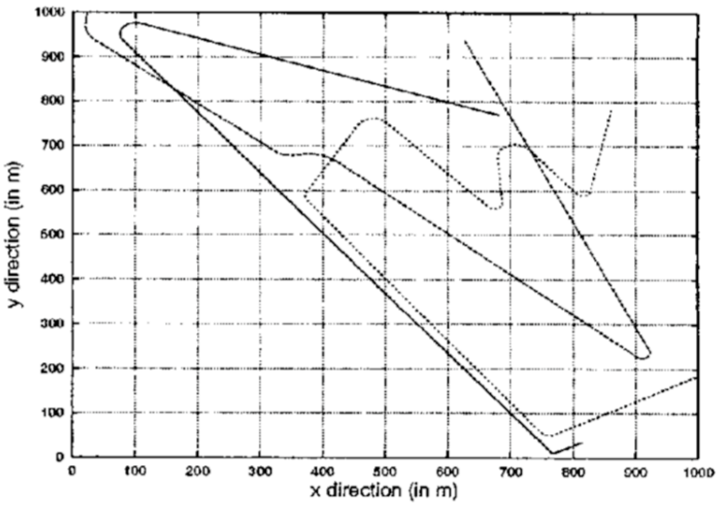


Figure 68: Trajectory of three nodes moving with the Smooth movement pattern (From [Beto1b]).

responsible for the speed selects a target speed from a set of predefined possible speeds. The node then accelerates (or decelerates) linearly until reaching the new target speed, with the acceleration limited in a given interval. When the other process chooses to change the direction, it select an angle in $[0, 2\pi)$. The turn is divided into N equal angles, where N is selected uniformly in $[2; 10]$, each part of the angle taking $1s$.

The probabilities with which the target speeds and the new directions are selected may vary in order to achieve different scenarios. It is also possible to choose correlated direction and speed change processes, thus simulating "stop-turn-go" behavior or slowing down of turning nodes [Beto1b].

A.2.1.1.2 RD WITH LOCATION DEPENDENT PARAMETRIZATION

The model proposed in [GSN05, GSN06] extends the RD model by allowing all random variables to be with location-dependent distribution. That is, the move direction, the speed, the pause time, and the segment length are all drawn from user-defined distributions, and these distributions may differ for the different zones of the simulation area, or even for every point (if the distributions are given as mathematical equations for example).

The model allows the creation of mobility patterns with user defined node densities by modifying the different distributions. It is a simple, tractable, synthetic model which presents more possibilities than the ordinary RD. However, it has not been sufficiently studied, and even tough it has been shown that it is capable of modeling the patterns of RWP, it is not clear if it is applicable to real-world scenarios. For example it is unable to model patterns where any of the distributions depends on the last visited location.

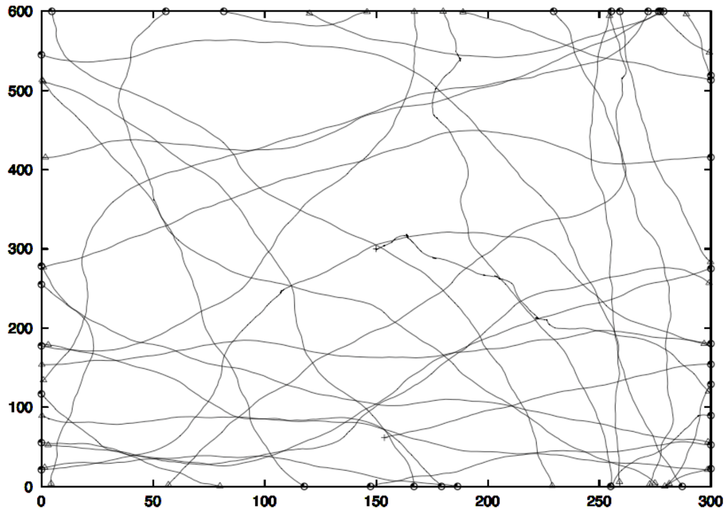


Figure 69: Trajectory of a node moving with the Boundless movement pattern in a 2D area (From [CBD02]).

A.2.1.2 *Boundless Simulation Area*

A movement window is selected, during which the node moves with the same speed in the same direction. At the end of this movement window, the speed and the direction are modified, with the modifications selected randomly with uniform distribution in given intervals. Upon reaching the border, the node is wrapped to the other side of the simulation area. Shown in Fig. 69.

A.2.1.3 *Graph-constrained*

A.2.1.3.1 CITY SECTION

In the *City Section* model introduced in [Dav00], the graph is a rectangular grid, where each intersection is also a vertex (the graph is planar) as illustrated in Fig. 70. The node chooses a destination vertex, and then follows the shortest path towards it, with at most one vertical and one horizontal movement. The speed is selected uniformly in a predefined interval. After reaching the destination, the node pauses for a randomly selected duration, which is also uniformly distributed in a given interval.

A.2.1.3.2 MANHATTAN

Also introduced in [BSH03], the *Manhattan* Mobility Model has the same restrictions on the speed as the Freeway Model. However, in this case nodes can change directions at the intersections with probability $p = 0.5$ to stay on the same street, and equal probabilities of $p = 0.25$ to turn left or right. The scenario is illustrated in Fig. 71. Both the Freeway and the Manhattan models initialize the nodes by placing them randomly on the lanes, and use an "erase-and-insert" border behavior, with the nodes trying to exit the simulation area being removed from the simulation, and a new node being inserted following the initialization rule.

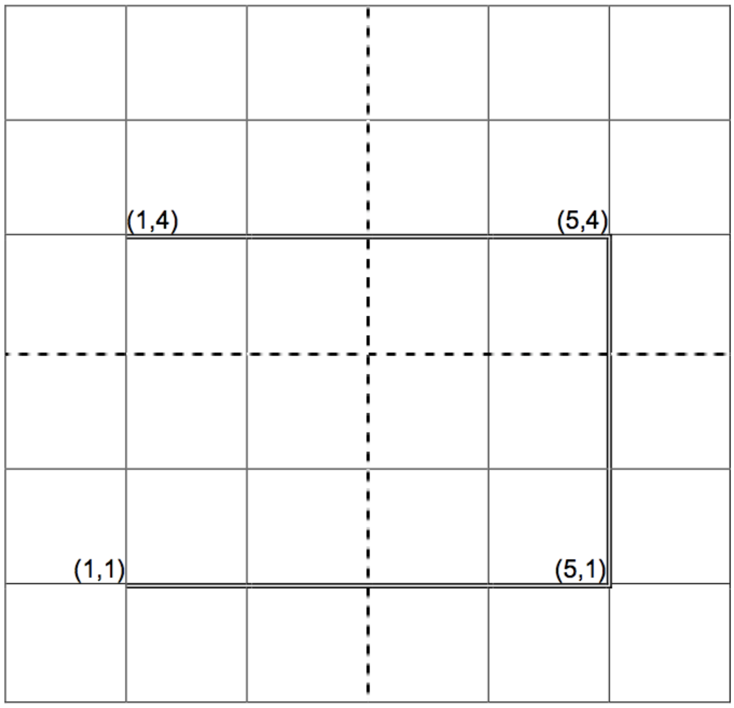


Figure 70: Trajectory of a node moving with the City Section movement pattern in a 2D area (From [Dav00]).

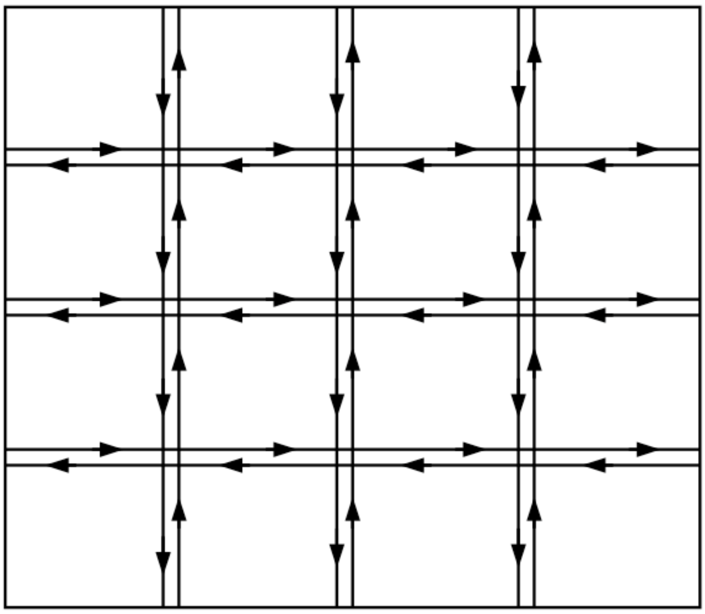


Figure 71: Trajectory of a node moving with the Manhattan movement pattern in a 2D area (From [BSH03]).

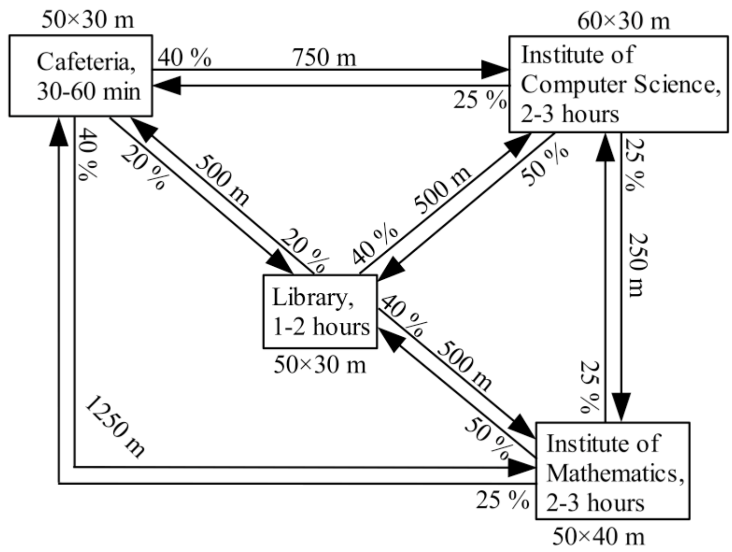


Figure 72: Planar graph of the Area Graph-based mobility model (From [BRS05]).

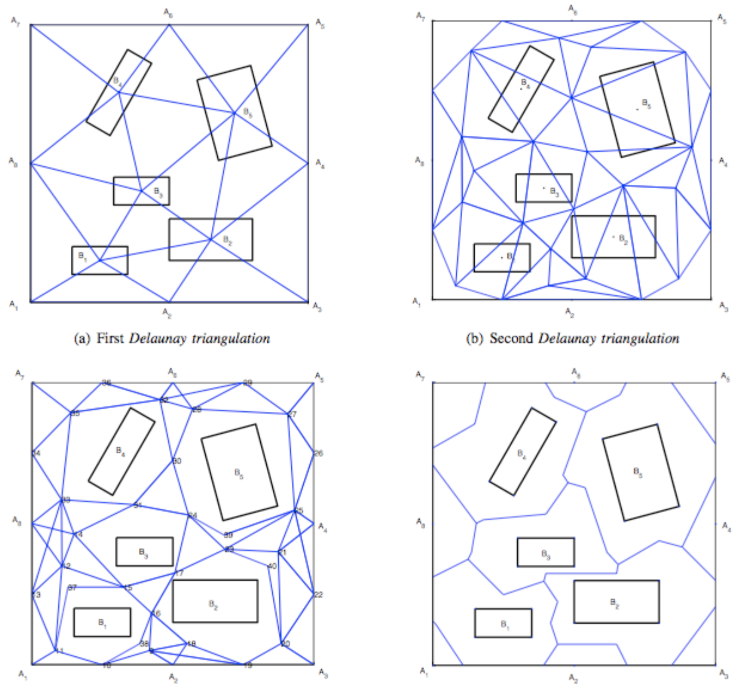
A.2.1.3.3 AREA

The Area Graph-based model [BRS05, BRS04] uses directed weighted graph to restrict node movement on the edges, and rectangular zones as vertices (Fig. 72). A node moves according to the RWP when in a vertex zone. The stay in such zone is chosen uniformly in an interval, each zone having its proper interval. After the stay time is up, the node leaves using an outgoing edge. An edge is selected with probability equal to its weight, that is, the weights of all outgoing edges of a given zone form a probability distribution from which is drawn the edge to be selected. The speed of the node is selected with uniform distribution in a given interval.

A.2.1.3.4 DELAUNAY

In [Hua05] the authors argue, that paths should not pass through obstacles and propose a way of constructing obstacle detour paths. A two-stage Delaunay triangulation is used to build the constraining planar graph (Fig. 73). In the first stage, they create a Delaunay triangulation of the simulation area by using the gravity centers of the buildings and the midpoints of the simulation border as triangulation points. Then, the second stage performs a new Delaunay triangulation, this time using the midpoints of the edges which were created during the first stage. Finally, all edges intersecting an obstacle are removed from the graph generated from the second-stage triangulation. The edge removal procedure is the following: if an edge crosses to parallel sides it is deleted, otherwise it is replaced by two edges which connect the vertices of the old edge with the midpoint between the closest obstacle edge and the closest graph vertex.

It was found that compared to a single stage Voronoi diagram generation, the proposed mechanism generates more detour paths. Furthermore, the shortest paths in the Delaunay model are shorter than detour



(a) First *Delaunay triangulation*

(b) Second *Delaunay triangulation*

(c) Detour paths constructed by trimming the second *Delaunay triangulation*

(d) Detour paths generated by *Voronoi model*

Figure 73: Using Delaunay triangulation and Voronoi diagram to construct detour paths (From [Hua05]).

paths generated by the Voronoi diagram. However, it is not clear if the same results would remain valid if a similar, two-stage procedure is also applied to the Voronoi diagram approach. Furthermore, the Voronoi edges were removed using a simpler procedure where an edge is removed if it intersects the obstacle.

A.2.1.4 *Group*

A.2.1.4.1 STRUCTURED GROUP

Another enhancement of the RPGM is the *Structured Group Mobility Model* [BL04]. There, the position of each node is not fixed to a reference point, but instead on each step its position with respect to the group center is selected from two distributions (given as parameters). First, the distance from the center is drawn, and then the angle with respect to the orientation of the group center. The authors have explicitly stated the possibility to recursively define group of groups, which although not specified for the RPGM does not present any technical difficulty.

This definition makes the model more flexible when compared to the RPGM, but also more difficult to configure. Furthermore, it shares the same drawbacks as the RPGM.

A.2.1.4.2 OTHER

There exist other group models, which are less frequently used.

One example is the *Exponential Correlated* model [HGPC99, CBD02] which uses an equation to specify the node position. By selecting close

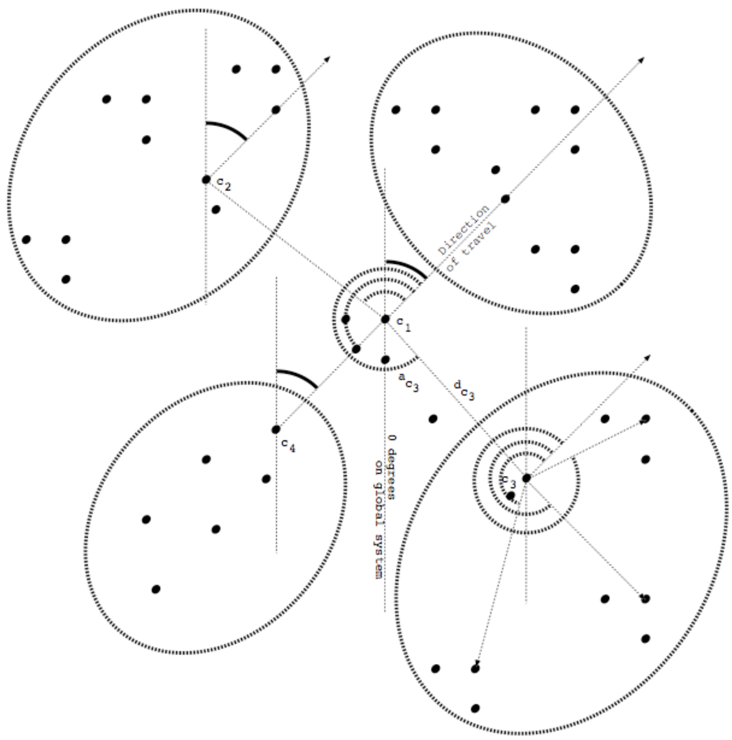


Figure 74: The position of all nodes of a group divided into four sub-groups moving with the Structured Group movement. (From [BL04]).

initial parameters one obtains nodes which have close trajectories, thus simulating a group behavior.

Another example is the *Virtual Track* [ZXG04] mobility model, which restrains the movement of the nodes on a set of virtual tracks connecting several switch stations. The only group membership criteria is that the nodes are moving in the same direction, with almost the same speed. Thus, when a group arrives at a switch station, if some of the nodes select different destination, the group splits. Conversely, if two groups arrive at the same time in a switch station, and select the same destination, they will merge into a new group.

A.3 ADDITIONAL EMPIRICAL AND DATA-DRIVEN MOBILITY MODELS

A.3.1 *Coarse-grained Trace Based Models*

A.3.1.1 *WLAN trace based*

A.3.1.1.1 TUDUCE ET *al.* [TG05]

The behavior of the nodes is modeled with active-inactive cycles (with respect to networking activity). During the active state, the node is connected to an AP while during the inactive state the node moves between APs. The authors argue that this corresponds to the typical usage of a real WLAN user - open device, work, close device, move. This, however, is valid only for laptop users, as VoIP phone users normaly keep their phones always on. A handover is simulated with an active-inactive-active cycle with a zero duration of the inactive period.

A.3.1.1.1.1 Data description Traces are gathered from 166 access points spread accross 32 buildings (no dormitories) in a university campus. Four traces are analysed, with data polling for the first two (19/May/2003-mid June; mid June-09/July/2003) and event-based data gathering for the second (April 2004; May 2004). The first two traces contain 3073 users, while the second two - 4762, where each unique MAC address is considered as a separate user.

A.3.1.1.1.2 Mobility model The model is defined in a 3D simulation area shaped as a rectangular parallelepiped. However, it may be applied to any other type of simulation area. Node movement generation is performed based on the following parameters:

- Maximal node speed - v_{max}.
- Intra-cell movement duration probability density - $P_i(x)$.
- Cell width - C_{width}.
- Number of cells for each of the dimensions - $(C_x; C_y; C_z)$.
- Maximum number of cells a node can visit - A_{max}.
- Distribution of the number of visited cells - $A(i)$.
- Probability to remain in the same cell - p_{same}.
- Probability to move to a neighboring cell - p_{neigh}.
- Probability to move to a non-neighboring cell - $p_{non-neigh}$.

Nodes are distributed uniformly in the simulation area. For each node is generated a list of accessible cells. Afterwards, the node selects uniformly from its accessible cells the destination cell towards which to move. Then, if the node has to stay in the same cell, the duration of stay is chosen from $P_i(x)$. If the node has to move to another cell, it passes in *inactive* mode and moves with a uniformly chosen speed in $(0, v_{max}]$. Then, the process repeats.

The list of accessible cells for a node is generated as follows. First, the number of accessible cells for this node is chosen from the distribution $A(i)$. The current cell of the node is added to the list (the one where it was put during the initialization). Then, cells are added one by one to the list, the cells being probabilistically selected. A cell is selected with probability p_{same} to be the same cell, with p_{neigh} to be a neighboring cell and $p_{non-neigh}$ to move to a non-neighboring one.

A.3.1.1.1.3 Parameter extraction The autors extracted the topology of the access points by processing the WLAN traces and declaring as neighbours each pair of APs for which there exist at least one roaming event in the event-based data. The observation that 64% of the users roam to a neighbouring AP on another floor justifies the decision to evaluate the model in a three dimensional space.

The coverege area of an AP (i.e. a zone, also refered as a "cell") is a cube, that could fit in a sphere with radius 175 m (which the authors measured as the maximal transmition range in open space) - i.e. $C_{width} = 200$ m.

A.3.1.1.1.4 Model validation The data were validated using the cross-validation method. In its simplest form, called holdout method, the data are split in two: a training set and a testing set. The model parameters are extracted only from the training set and the traces resulting from the execution of the model are compared to the data from the testing set. The error is estimated with the Root Mean Square Error (RMSE) aggregated over all points. The inter-cell movement time and the session lengths were well-approximated by the model.

Furthermore a parameter space investigation was performed. The metrics used to characterize the influence of the parameters are the average speed and the average relative speed. The parameters p_{same}, p_{neigh} and $p_{non-neigh}$ are not independent, so their influence is investigated simultaneously. The measurements show that for $p_{same} = 1$ the average speed is low, for $p_{neigh} = 1$ it increases and reaches its peak at $p_{non-neigh} = 1$. It should be noted that the model suffers from speed decay and requires a relatively long period to stabilize (3000s-4000s).

A.3.1.1.1.5 Discussion This is one of the few models which are explicitly defined for a 3D simulation space. Its macro-level movement characteristics have been validated by using cross-validation. The micro-level characteristics however were not studied. The authors did not point out how they processed the traces, e.g. if they addressed the ping-pong phenomenon. Surprisingly, the exact micro-level movement generation was not supplied. It is not clear how a node chooses the destination way point in the cell, neither how it moves toward it. We assume that the process follows the RWP approach because it is given as example in the paper, but this remains a speculation.

A.3.1.1.2 STATISTICAL [YNLK06]

A.3.1.1.2.1 Data description The parameters are extracted from Dartmouth College syslog traces.

A.3.1.1.2.2 Data preparation The traces are treated on a building-only level, i.e. all APs belonging to the same building are coalesced to a single location. First, the traces are processed in order to smooth out the pinpong phenomenon. The suggested smoothing algorithm calculates for each new association an "average position" for the user, within a time window, by using her registered positions in the past, weighted by their respective stay times. Afterwards, the AP closest to the "average position" is selected.

For each user, her stay time at an AP (or her stay time at the same AP averaged over the whole trace) is used to infer if the AP is a *stationary* or *transient* point. The *stationary* points are considered as trip endpoints, while the *transient* points are used to refine the trip trajectory. The authors analyzed the histogram of the stay times and fixed that a *transient* point is a point with stay time shorter than 3 mins.

A.3.1.1.2.3 Mobility model In this model, node movements are constrained to undirected graph given as parameter. A path is defined as a set of intermediate vertices to be visited between the start and end points. The model generates node movement with the help of the following parameters:

- Initial node distribution.
- An OD matrix.
- Minimal and maximal speed.
- Pause time per building.
- Path selection probability.
- Undirected graph.

In the beginning, the nodes are distributed according to the *Initial node distribution*. Then, a node selects the next building to move to by using the probabilities of the OD matrix. The path to be taken between the two buildings is then selected by following the probability distribution given in the *Path selection probability*. The node then moves at constant speed drawn uniformly from $[v_{min}, v_{max}]$ (the minimal and maximal speeds) by following the selected path. After reaching the destination building, the node pauses with the pause time provided for each building.

A note about the path selection probability should be made. The authors implemented it with a second-order Markov chain by turning the route probabilities into a set of localized, turn probabilities. The turn probability indicates in which direction will a node turn after reaching a vertex. It is defined by the following conditional probability:

$$Prob\,[next \mid current,\ previous,\ origin,\ destination]$$

where all terms refer to an intersection or building location. The two definitions are equivalent.

A.3.1.1.2.4 Parameter extraction The spatial constraints are extracted from a map of the campus. The constraints are modeled with a undirected graph, where the vertices represent buildings or intersections, and the edges - roads or paths. A curved road is approximated by a sequence of line segments and each connecting point is considered as an intersection. The weight of an edge corresponds to the length of the road. In addition, a *usage frequency* is assigned to each route.

Generally there exist many possible routes between two given locations. In this model route selection does not rely on a single (shortest) path - instead, a set of route candidates is defined. The set contains the a fixed number of reasonably long shortest paths (a path is reasonably long if it is shorter than C×*length-of-shortest-path*, where C is configurable). Choosing a particular route out from the route candidates is done in a random manner, where the probability of each route being selected depends on its usage frequency.

The frequency of each route is calculated before the model is executed. The trace data is processed and for each *trip* the frequencies of the routes involved are updated by the following algorithm. If the trace contains entries for *transient* points for this trip, the route closest to all those points is selected and its frequency is incremented by one. If not, the frequencies of all route candidates are incremented proportionally to the inverse of their length (i.e. shorter routes are incremented by more). More precicely, the frequency of the i_{th} route is incremented by Δ_i, given by $\Delta_i = \frac{\frac{1}{w_i}}{\sum_{k=1}^{N} \frac{1}{w_k}}$ where w_i is its weight, and N is the total number of route candidates generated for this origin-destination pair. Note that the sum of the frequencies of all candidates is equal to 1.

The initial user distribution is determined from the route frequency statistics by counting the frequency of visits to buildings when they occur as *stationary* points in the trace data. These statistics give the likelihood of selecting a certain building as an origin. The destionation distribution for each origin building is found by counting the relative frequency of each building that appears as a destination.

A.3.1.1.2.5 Model validation For the evaluation of the model, the authors manually counted the number of pedestrians passing through several intersections on the campus of Dartmouth College for a total of 7 hours over a period of two days, and compared these real measurements to the results produced by their trace-driven model. The traces of laptop users and VoIP telephone users are processed separately, because the latter generate much more association events and are presumed to be modeled more realisticaly. The correlation between the real measurements and the synthetic results at the intersections ranges from 0.7476 to 0.9270 for laptop users and from 0.8046 to 0.8305 for VoIP phone users. On average, the correlations of both types of users are greater than 0.8, implying that the real measurements and the generated results are very close. The RMSE indicated an average error of about 20%. Contrary to the expectations, the VoIP phone user traces gave worse results than the laptop user traces. The authors suggest that this is either due to the noise in the process, or due to the much smaller group of VoIP phone users (less than 5% of the laptop users), which makes it less representative as a whole.

Finally, the authors investigated the parameter space of their model with a two-month trace. The evaluation of the parameters affecting the route candidates set (N - maximal number of shortest paths to include and C - coefficient limiting the reasonable length of the candidates) shows that no matter what N and C are, the shortest route is taken from nearly 40% to 60% of the time, and the top five shortest routes are nearly always taken, with a negligible selection of the rest of the routes. They also point out that less than 10% of the laptop trips have *transient* points, versus more than 30% for the VoIP trips. Unlike the route selection, the user density on pathways seems to change more importantly with the variance of N and C.

A.3.1.1.2.6 Discussion The paper provides an interesting approach to validating the generated traces - counting the number of people crossing a given point and comparing it to the simulated results. However, no other metrics with known realistic values were used for evaluation. For example it would be helpful to count not only the individuals passing through the observable points, but also the formed groups as in [BRC+04]. Furthermore, the selected metric is very coarse and it is not clear if a purely synthetic model would not also fit. That is a general parameter exploration remark - even though the effects of the selection of the shortest paths were studied, the impact of the other parameters (most notably the OD-matrix and the pause time distribution) was not analyzed, so it is not known if the validity of the model is due to the entire modeling process, or due to the environmental graph for example.

A.3.1.2 *Activity-based*

The activity-based models are founded on the principle, that a trip is a derived demand, being a means to an end, rather than an end of itself [SK99a]. An activity is a collection of actions typical for the considered scenario [BKOKR04].

Here, we are providing the way activity modeling was implemented in [SK99a,SK99b], but it was also used as a foundation of other mobility models, e.g. [Ste02, Ray03, BKOKR04, CB05, KB05, ZHL06, KRKB06]. However, even if there may be some differences in the way activity is defined, or the accompanying characteristics implemented, the core of the approach remains unchanged. A survey of the activity modeling techniques may be found in [AG92].

A.3.1.2.1 DATA DESCRIPTION

The activity transition and duration matrices used by this mobility model were derived from a trip survey [Ass89] where a travel diary was completed by each household member over 5 years of age, in which details on all trips taken during the survey day were recorded. Each recorded trip included the trip start and end times, the trip purpose at the origin and destination, and employment status.

Trips are classified in nine categories: work, work-related, school, serve passenger, shopping, social/recreation, personal business, return home and other. The day is divided into twelve equal time periods which are used to aggregate the data from the survey. The authors observed that there exist several types of behavior and decided to define four categories of persons:

1. Full-time employed outside the house
2. Part-time employed outside the house, but not student
3. Student, secondary or post-secondary, possibly employed part-time outside the home
4. Not employed outside the home, and not a student

Each category possesses its own parameter values. Hereafter, the description of the parameters and the algorithms are given for a single user set. The existence of several user sets can be achieved by running simultaneously several such "simplified" models, which differ in their userset-dependent parameters.

A.3.1.2.2 MOBILITY MODEL

The following parameters are used for mobility generation:

- Mean speed.
- Activity transition matrix.
- Activity duration matrix.
- Per-activity zone weight.
- List of intermediate zones to be visited when moving between two zones.

The simulation area is a 2D rectangle, divided into equaly-sized zones. A zone may have weight associated to it for each of the activities, although in the article only zones with shopping activity have associated weights (representing the zonal retail employment). An activity is modeled as the triplet $(timeofday, duration, zone)$. In the beginning of the simulation, each node is assigned a current activity (e.g. home) and typical zones for some of its activities (e.g. home, school and work zones).

The model is then executed by first selecting the next activity to be carried out A and its duration. The activity is drawn randomly from the activity transition matrix, and depends on the current activity and time. The duration T_A of this activity is chosen from a distribution depending on the next activity A and the current time. Then, if there is a typical zone for the activity, it is selected as the destination zone. Else, if there are zones with weight associated for this activity, the destination zone is selected randomly from the top five, after they were ordered in a descending order on their weight divided by the distance to them. Otherwise, the destination zone is randomly selected. Once the destination zone is selected, a trip is created from the list giving the zones to be visited when going from the current to the destination zone.

A.3.1.2.3 DISCUSSION

The approach of activity-modeling provides many opportunities to realistic mobility modeling. However, it requires a detailed activity list for a large population - data which is particularly difficult to obtain. Additionally, it may only be used to model existing human societies, which means that making predictions for future scenarios, or simulating other types of entities (e.g. animals) is not going to be straightforward.

However, if activity data is available, it may provide realistic macro-level behavior. Additionally, based on the activities, one may infer other types of information regarding the simulation scenario which could lead to more realistic simulations as a whole. For example, the simulation creator may assign different classes of data traffic to the different activities.

A drawback of this approach is that in general it requires a lot of background information and a lengthy algorithm description in order to detail the simulation scenario. For that reason most of the activity-based mobility models were not validated, or validated very briefly. The lack of such validation, together with the complexity of the parameters have a serious negative impact on the acceptance of such models.

A.3.1.3 *Survey-based*

The survey-based models attempt to model real user behavior by extracting their parameters from surveys filled by volunteers. Because the surveys generaly contain questions on the purpose of stay, a network usage model can also be defined.However, it is not clear how such models can be validated if only the information from the surveys is used. This suggests that in order to prove that the movement patterns obtained by such model match some real-world characteristic, some other methods of data collection should be used (e.g. WLAN traces).

We are going to present the Weighted Waypoint Mobility Model [HMS+04, HMS+05] as an example.

A.3.1.3.1 DATA DESCRIPTION

The authors extracted the parameters for their model from a mobility survey targeted at randomly selected students on the campus of the University of Southern California, USA. It is with a per-building granularity, asking the students for their previous, current and next location as well as the stay at each of these three buildings. The locations were then aggregated by the types of zones and the time period (morning or afternoon) - e.g. the number of all students in the library intending to go to the cafeteria in the morning. The pause times were grouped only by the types of zones.

A.3.1.3.2 MOBILITY MODEL

Several zones are created in the simulation area (corresponding to buildings). A transition matrix, given as parameter, provides the probability with which is going to be selected the next building. The transition matrix depends on the current building and the current time of day. After the building is selected, a point is randomly picked in that building, and the node moves at a speed uniformly distributed in a provided range. Once the destination is reached, the node pauses, with pause time drawn from a distribution which depends on the destination zone.

A.3.1.3.3 PARAMETER EXTRACTION

Based on its specifics the simulation zones are classified into 5 types - 3 types of buildings (classrooms, libraries and cafeterias), other buildings and off-campus area. The analysis and the tuning of the model were based on a total of 268 responses, collected in a $3\frac{1}{2}$ week period. The pause time distribution type varies for the different zone types - in classrooms it is like a bell-shaped normal distribution while in libraries it is a heavy-tail distribution.

A.3.1.3.4 VALIDATION

This model shows uneven spatial distribution of the nodes, as they tend to cluster in popular zones and stay longer there. Since the transition matrix is time-dependent, the distribution of the nodes never reaches a steady state, which as the authors underline, suggests that converging to a steady-state distribution is not necessarily a requirement of realistic mobility models. The results obtained with this model, however, were not validated, e.g. the model is accepted as a-priori realistic and no in-depth analysis of its mobility characteristics was performed.

A.4 PROOFS OF LEMMAS AND THEOREMS

A.4.1 *Proof of Lemma 5.2.3*

Proof. First, we should note that the measurable functions in Γ are $\mathcal{F} \otimes \mathcal{E}/\mathcal{E}^{[0;1]}$-measurable, and the measurable functions in Δ are $\mathcal{F} \otimes \mathcal{E}/\mathcal{T}^{[0;1]}$-measurable.

For every given starting position e_0 we can then define the set of visited endpoints $E_\Gamma(e_0) = \{e_i | i \in \mathbb{N}\}$ as:

$$\begin{aligned}
&e_0 := e_0\\
&e_1 := \gamma_1(\omega, e_0)(1)\\
&\dots\\
&e_i := \gamma_i(\omega, e_{i-1})(1)\\
&\dots
\end{aligned}$$

and for every starting time t_0 the set of endpoint visit times $T_{\Delta,\Gamma,t_0,e_0} := T_\Delta(t_0, E_\Gamma(e_0)) := \{t_i | i \in \mathbb{N}\}$ as:

$$\begin{aligned}
&t_0 := t_0\\
&t_1 := \delta_1(\omega, e_0)(1) + t_0\\
&\dots\\
&t_i := \delta_i(\omega, e_{i-1})(1) + t_{i-1}\\
&\dots
\end{aligned}$$

Let $i(t)$ be the index of the smallest t_i bigger than t, that is $i(t) := \{i | t_i = \min\{T_{\Delta,\Gamma,t_0,e_0} > t\}\}$. Then, the function

$$f_{\Gamma,\Delta}(\omega, t, e_0, t_0) := \gamma_{i(t)}(\omega, e_{i(t)-1})\left((\delta_{i(t)}(\omega, e_{i(t)-1}))^{-1}(t)\right)$$

is defined for all $t > t_0$. The function f depends on the initial conditions (t_0, e_0) and is limited on the left, i.e. it is undefined when $t \leqslant t_0$. We can select a function f^* defined over the whole T as follows:

$$f^*_{\Gamma,\Delta}(\omega, t, e_0, t_0) = \begin{cases} f_{\Gamma,\Delta}(\omega, t, e_0, t_0), t > t_0 \\ [t \to e_0], t \leqslant t_0. \end{cases}$$

Now, we can define the mobility model $m_{\Gamma,\Delta} : \Omega \times E \times T \to E^T$ as:

$$m_{\Gamma,\Delta} : (\omega, e_0, t_0) \to [t \to f^*_{\Gamma,\Delta}(\omega, t, e_0, t_0)]$$

Indeed, $f_{\Gamma,\Delta} : \Omega \times T \times E \times T \to E$ is $\mathcal{F} \otimes \mathcal{E} \otimes \mathcal{T}/\mathcal{E}^T$-measurable in each interval $(t_{i(t)-1}; t_{i(t)}]$ as in this interval it is the composition of two measurable functions :

1. $\gamma_{i(t)}$, which is measurable by definition, and
2. $\delta^{-1}_{i(t)}$, as the inverse function of a measurable function.

However, this means that $f_{\Gamma,\Delta}$ is $\mathcal{F} \otimes \mathcal{E} \otimes \mathcal{T}/\mathcal{E}^T$-measurable for the interval $(t_0, \sup(T))$ as it is a countable union of disjoint $\mathcal{T}$-measurable intervals on which f is measurable. On the other hand $f^*_{\Gamma,\Delta}$ is measurable for $(\inf(T); t_0]$ as it is a constant function, which provides the measurability of $m_{\Gamma,\Delta}$.

□

A.4.2 *Proof of Theorem 5.2.2*

Proof. The basic idea is to recursively define the mobility model for each set of parameters of the different layer functions. Then start from an initial configuration (such as position in the space and time) and following the recursive definition provide the trajectory for an individual node. We are going to prove the theorem in the case where there is a stay layer function - the proof is analogical for the case where there is no stay layer.

Lets set the sample space $\Omega := \Omega_{str} \times \Omega_{map} \times \Omega_{tac} \times \Omega_{dyn} \times \Omega_{sta} \times \Omega_{init}$, and let $\omega = (\omega_{str}, \omega_{map}, \omega_{dyn}, \omega_{sta}, \omega_{init}) \in \Omega$. The space Ω_{init} specifies the initial conditions of the movement. For simplicity, we will carry out our proof in the case when $\Omega_{init} = T \times E$ (e.g. we have the start time of the traces t_{init} and the initial position of the node e_{init}), but the same construction applies for any other initialization rule that uniquely identifies a trajectory (e.g. one may specify the current route, the time remaining until reaching its end, and if the next process should be a stay). We set $\omega_{init} = (t_{init}, e_{init})$ and assume that the node will first move to a zone.

In order to simplify the proof we introduce the following notation (related to the upper three LEMMA layers):

$$
\begin{aligned}
(Z^n_{str}, T^n_{str}) &:= l_{str}(\omega_{str}, n) \\
e^n &:= l_{map}(\omega_{map}, n, Z^n_{str}) \\
g^n_{tac}(\omega, e) &:= l_{tac}(\omega_{tac}, n, e, e^n)
\end{aligned}
$$

If we now denote the movement functions (corresponding to the lower two LEMMA layers) as:

$$
\begin{aligned}
\left(g^n_{dyn}, d^n_{dyn}\right) &: (\omega, e) \to l_{dyn}(\omega_{dyn}, n, g^n_{tac}(\omega, e)) \\
(g^n_{sta}, d^n_{sta}) &: (\omega, e) \to l_{sta}(\omega_{sta}, n, T^n_{str}, Z^n_{str})
\end{aligned}
$$

we can construct the sets Γ and Δ as defined in 5.2.3:

$$
\begin{aligned}
\gamma_{2i+1} &:= g^i_{dyn} \\
\delta_{2i+1} &:= d^i_{dyn} \\
\gamma_{2i} &:= g^{i-1}_{sta} \\
\delta_{2i} &:= d^{i-1}_{sta}
\end{aligned}
$$

for $i \in \mathbb{N}_+$. Indeed, these functions are all measurable in the sense required by the lemma (by definition of the layer functions), which means that following the construct given in 5.2.3, based on the layer functions $l_{idx}, idx \in \{str, map, tac, dyn, sta\}$, we can create a mobility model f_m as defined in 5.1.2, as:

$$f_m : \Omega \times E \times T \times T \to E$$

□

A.4.3 *Proof of Theorem 5.4.1*

Proof. We will now prove the theorem by constructing the required layer functions with the help of the model m. Without loss of generality, we are going to fix $\Omega_{idx} := \Omega$. The path corresponding to $\omega \in \Omega$ is $m(\omega, t)$ and may be regarded as such both when m is a deterministic function or a stochastic process.

We will start the construction by first defining some auxiliary constructs. Starting from a time t_0, we define the sequence $T^m(\omega, t_0) := \{t_i | i \in \mathbb{N}\}$ as the times at which the sample path enters a new zone,

$T^m_{stay}(\omega, t_0) := \{t_i | i \in \mathbb{N}\}$ as the stay time in that zone, and $Z^m(\omega, t_0) := \{Z_i | i \in \mathbb{N}\}$ as the sequence of these zones.

To define this more formally, lets denote all zones containing the point e with $Z(e)$, and $\bar{Z}(e) = \mathcal{Z} - Z(e)$. Lets also denote the function returning the next time instant in which the sample path reaches a new zone $\tau^m(\omega, t_i, \mathfrak{z}) := \inf(t) : t > t_i, m(\omega, t) \in \mathfrak{z}$, and the duration of the stay in that zone $\tau^m_{stay}(\omega, t_i, \mathfrak{z}) := \inf(t) - \tau^m(\omega, t_i, \mathfrak{z}) : t > \tau^m(\omega, t_i, \mathfrak{z}), m(\omega, t) \notin \mathfrak{z}$. Then, we define T^m as:

$$
\begin{aligned}
t_0 &:= t_0, \\
t_1 &:= \tau^m(\omega, t_0, \cup \bar{Z}(g_\omega(t_0))), \\
&\dots \\
t_i &:= \tau^m(\omega, t_{i-1}, \cup \bar{Z}(g_\omega(t_{i-1}))), \\
&\dots,
\end{aligned}
$$

the sequence of zones as:

$$
\begin{aligned}
z_0 &:= Z(m(\omega, t_0)), \\
z_1 &:= Z(m(\omega, t_1)), \\
&\dots \\
z_i &:= Z(m(\omega, t_i)), \\
&\dots,
\end{aligned}
$$

and the sequence of stay times as:

$$
\begin{aligned}
t_0^{stay} &:= \tau^m_{stay}(\omega, t_0, z_0), \\
t_1^{stay} &:= \tau^m_{stay}(\omega, t_1, z_1), \\
&\dots \\
t_i^{stay} &:= \tau^m_{stay}(\omega, t_i, z_i), \\
&\dots
\end{aligned}
$$

Given the initialization values t_{init} and e_{init} we can now construct the different pre-layer functions as follows:

(i) Constructing a strategy pre-layer function.

If we denote with l^m_{str} the strategy function induced by m, then:

$$l^m_{pre-str} : (\omega, n) \to (z_n, t_n^{stay})$$

In other words, $l^m_{pre-str}$ selects the next zone to be visited by the node by tracking the function m.

(ii) Constructing a mapper pre-layer function.

The mapper layer function, induced by m is:

$$l^m_{pre-map} : (\omega, Z, n) \to m(\omega, t_n)$$

The mapper projects every zone (given as parameter from the strategy) as the first point of that zone to be reached by the function m.

(iii) Constructing a tactic pre-layer function.

The tactic pre-layer function, induced by m is:

$$l^{m}_{pre-tac} : (\omega, n, e_s, e_e) \rightarrow$$
$$[x \rightarrow m(\omega, x * (t_n - t_{n-1} - t^{stay}_{n-1}) + t_{n-1} + t^{stay}_{n-1})]$$

The tactic maps the parameters to a function in $\mathcal{G}$ following the path of m by also taking into account the effects of the stay pre-layer function.

(iv) Constructing a dynamic pre-layer function.

The dynamic pre-layer function, induced by m is:

$$l^{m}_{pre-dyn} : (\omega, n, g) \rightarrow$$
$$(g, [x \rightarrow x * (t_n - t_{n-1} - t^{stay}_{n-1}) + t_{n-1} + t^{stay}_{n-1}])$$

As defined, the dynamic maps the given function to a vector in $\mathcal{G} \times \mathcal{D}$ as generated by m.

(iv) Constructing a stay pre-layer function.

The stay pre-layer function, induced by m is $l^{m}_{pre-sta}$:

$$l^{m}_{pre-sta} : (\omega, n, t_{stay}, z_{stay}) \rightarrow$$
$$(x \rightarrow m(\omega, x * t^{stay}_{n} + t_n),$$
$$[x \rightarrow x * t^{stay}_{n} + t_n])$$

These pre-layer functions, by definition, coincide with m when looked at the different levels (e.g. as a strategy, as a dynamic). However, they cannot be used with another layers, as each layer is adapted to the behavior of the other layers (for example the mapper expects the destination zones to be given in a specific order).

We can generalize the m-induced pre-layer functions to full layer functions aid of the helper functions F defined hereafter. First, let $F^{Z_i}_{Z_j} : Z_i \rightarrow Z_j, Z_i, Z_j \in \mathcal{Z}$ are a set of homeomorphic functions, such as $F^{Z_i}_{Z_i} := id$ (that is, these functions preserve the points of each of the zones when the zones are projected into themselves). Lets now define the function $F : \mathcal{Z} \times \mathcal{Z} \rightarrow \mathcal{Z}$ as $F(Z_i, Z_j) := F^{Z_i}_{Z_i}$. Note, that this requirement puts a constraint on the zones $\mathcal{Z}$. However, this is only a mild restriction and does not limit the generality of the solution. Further, let $H := H(e^1_s, e^1_e, e^2_s, e^2_e) : E \rightarrow E$ be a homeomorphism, such as $H(e^1_s) = e^2_s$ and $H(e^1_e) = e^2_e$.

Armed with all the necessary pre-layer functions and helper functions, we can build the full m-induced layer functions as follows:

(i) Constructing a strategy layer function.

The strategy has no need of adaptation, as by design it is the highest layer and thus determines the behavior of the other layers. Thus,

$$l^{m}_{str} \equiv l^{m}_{pre-str}$$

(ii) Constructing a mapper layer function.

$l^{m,F}_{map} : (\omega, Z, n) \rightarrow F\left(Z, l^{m}_{pre-map}(\omega, t, Z)\right)$

(iii) Constructing a tactic layer function.

The tactic layer function, induced by m is: If we denote the function returned by the pre-layer function as: $g = l^{m}_{pre-tac}(\omega, n, e_s, e_e)$, then:

$$l^{m,H}_{tac} : (\omega, n, e_s, e_e) \to$$
$$[x \to H(e_s, e_e, g(0), g(1))(g(x))]$$

(iv) Constructing a dynamic layer function.

The dynamic layer function can also remain unchanged when compared to the pre-layer version, as it does not change the trajectory and only adds the time of execution. Thus,

$$l^{m}_{dyn} \equiv l^{m}_{pre-dyn}$$

(iv) Constructing a stay layer function.

If we denote the functions returned by the pre-layer function as: $(g, d) = l^{m}_{pre-sta}(\omega, n, t_{stay}, z_{stay})$, then:

$$l^{m,F}_{sta} : (\omega, n, t_{stay}, z_{stay}) \to$$
$$\left([x \to F(z_{stay}, g(x))], \left[x \to t_{stay} * \frac{d(x) - d(0)}{d(1) - d(0)}\right]\right)$$

These layer functions are induced by m and each of them follows exactly the paths generated by m whenever the upper layers behave as expected. Otherwise, the path is adjusted with the help of the homeomorphic functions F, which preserves some of the properties of m (e.g. if m is a stationary stochastic process, then the resulting process is also stationary). □

A.4.4 *Proof of Theorem 5.5.1*

Proof. We are going to use the following variants of the layer processes (modifications of the original ones). Furthermore, without loss of generality, we will assume that all layer processes are defined on the same space Ω. Finally, we will be using the *random functions* corresponding to the stochastic process layers defined in 5.3.1, which we will denote with L followed by the appropriate indexed. This equivalence, along with the necessary space transformations, are given in [SK].

- Modification of the strategy layer process:

$$L'_{str,z} : \Omega \to \mathcal{Z}^{\mathbb{N}} \tag{A.1}$$

$$L'_{str,t} : \Omega \to T^{\mathbb{N}} \tag{A.2}$$

- Modification of the mapper layer process:

$$L'_{map} : \Omega \to (\mathfrak{Z}^{\mathcal{Z}})^{\mathbb{N}} \tag{A.3}$$

- Modification of the tactic layer process:

$$L'_{tac} : \Omega \to (\mathcal{G}^{3\times 3})^{\mathbb{N}} \tag{A.4}$$

- Modification of the dynamic layer process:

$$L'_{dyn} : \Omega \to ((\mathcal{G} \times \mathcal{D})^{\mathcal{G}})^{\mathbb{N}} \tag{A.5}$$

Using these modifications we can express the compositions of the layers as follows:

$$L'_{ms} := L'_{map} \circ L'_{str,z} \tag{A.6}$$

$$L'_{tms} := L'_{tac} \circ L'_{ms} \tag{A.7}$$

$$L'_{dtms} := L'_{dyn} \circ L'_{tms} \tag{A.8}$$

The above compositions may be expressed with the help of the operators $f_2(f, a) := f(a)$ and $f_3(f, a, \theta) := f(a, \theta a)$:

$$L'_{ms}(\omega, n) := f_2\Big(L'_{map}(\omega, n), L'_{str,z}(\omega, n)\Big) \tag{A.9}$$

$$L'_{tms}(\omega, n) := f_3\Big(L'_{tac}(\omega, n), L'_{ms}(\omega, n), \theta\Big) \tag{A.10}$$

$$L'_{dtms}(\omega, n) := f_2\Big(L'_{dyn}(\omega, n), L'_{tms}(\omega, n)\Big) \tag{A.11}$$

where θ is a measure-preserving shift operator which transforms $X = (X_n)$ into $\theta X = (X_{n+1})$.

From lemma 5.5.2 follows that L'_{ms} is stationary, because L'_{map} and $L'_{str,z}$ are such. From lemma 5.5.3 follows that L'_{tms} is stationary, because L'_{tac} and L'_{ms} are stationary, and θ is a measure-preserving shift operator. L'_{dtms} is stationary because L'_{dyn} and L'_{tms} are stationary, and because of lemma 5.5.2.

This provides us with a process choosing the trajectory segments and their duration, which possesses a stationary regime. The finiteness of the trajectories and the movement durations provides the stationarity of m.

□

SIMULATOR

B

In this chapter we are going to describe the way we implemented the layered architecture and its capabilities. We have kept the spirit of the theoretical framework - simple, practical and powerful. The simulator is implemented by using SimPy [sim], an open-source, object-oriented, process-based discrete-event simulation language based on standard Python.

Python allows the so called *duck typing*, in which an object's current set of methods and properties determines the valid semantics, rather than its inheritance from a particular class [Wik08]. For that reason, the UML diagrams we have provided are not "strict", in the sense that an object is not required to inherit from the specified interfaces, but simply implement their methods. For example, we allow the usage of objects or functions interchangeably whenever possible (e.g. some of the layers are implemented as single functions).

Processes in SimPy are modeled with the help of *generator functions*. A generator function is a special type of function, which may return multiple values, suspending its execution after each returned value. It can be considered as an easy way to implement call-back functions [SPH]. We are not going to discuss the way these functions are implemented, but we are going to underline, that a generator function does not *return* a single value, but *yield*s more than once values to its caller. For that reason, we have indicated on the UML diagrams when a function yields values, and the types of these values.

B.1 ENVIRONMENT MODULES

There are three basic node environment modules - Environment, SimulationArea and Zone. Their relationships are displayed in fig. 75. We have taken a pragmatic approach to implementing the whole architecture and we have supplied only a 3D simulation area with rectangular zones. Any kinds of zones or simulation areas may be easily added. We are using NetworkX [net] as implementation for the graph constraints.

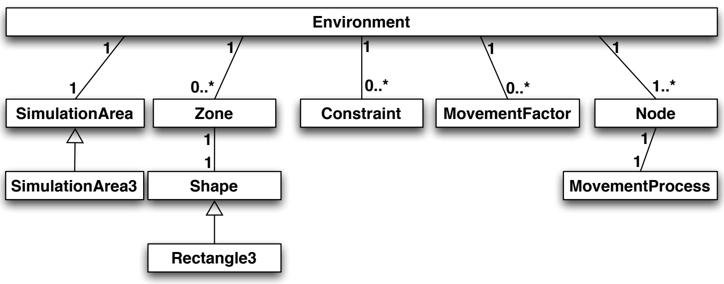


Figure 75: Relationships of the node environment objects.

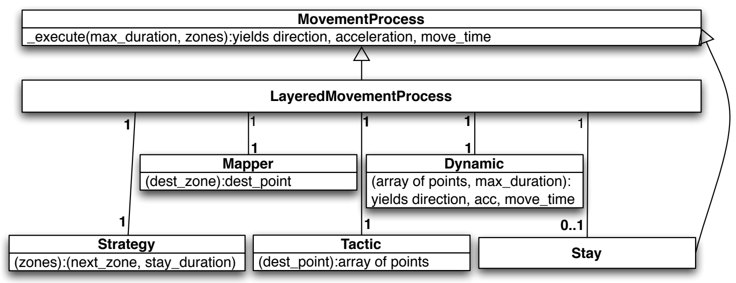


Figure 76: Relationships of the movement related elements.

B.2 MOVEMENT PROCESS

Each movement process is attached to a node, thus making the framework a multiagent microsimulator. The process implementation *yield*s node's new move direction (a 3D vector), acceleration (in m/s^2) and move duration (in seconds). The node moves for the specified time in the specified direction and acceleration, and after the time is up - the process is resumed to *yield* new values, repeating this process until the end of the simulation. It is possible to implement a movement process by inheriting directly from the MovementProcess class and overriding its *_execute* method (e.g. for an example, look the *lemma.layers.stay.pause.PauseStay*).

The LEMMA architecture is implemented by the LayeredMovementProcess (shown in fig. 76). The various layers are specified in the figure. The different layers have distinct, well defined functions. Our theoretical framework stipulates that the different layers need separate, distinct parts of the environment, e.g. the strategy would need only the zones, the tactic only the constraints etc. However, layer architects are not restricted to access any of these elements. Each of the layers can access the entire environment, along with the node which movement it governs. To do so, a layer has to define a "node" or "environment" member attribute, and it will be automatically set before the execution of the layer.

A layer implementation should be a function, or a callable object (i.e. defining a method with the special name *__call__*).

- Strategy - takes a list of zones as input, and outputs a tuple containing the destination zone and the stay time.
- Mapper - takes the destination zone as input, and outputs the destination point.
- Tactic - takes the destination point as input, and outputs a list of points to be visited (trajectory), ending with the destination point.
- Dynamic - takes the trajectory, and *yeild*s multiple movement direction/acceleration changes and durations, following the trajectory towards the destination point.
- Stay - takes the destination zone and the time to stay into it as input, and *yield*s movement direction and duration, keeping the node in its zone.

You can see the specification of these functions or callable objects in fig. 76.

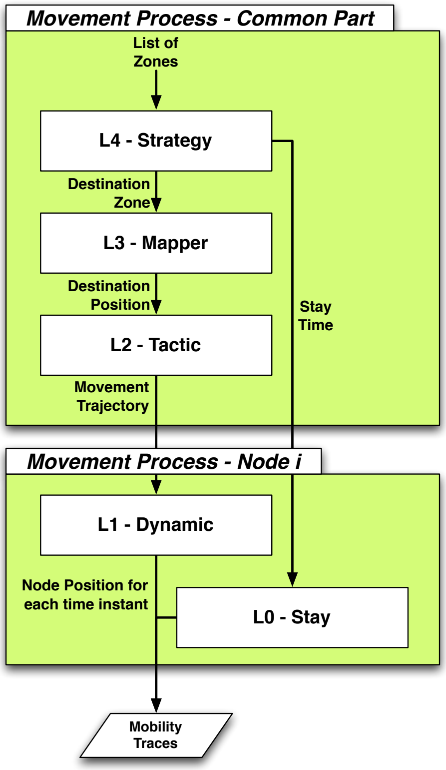


Figure 77: Group modeling principle.

B.3 GROUP MODELS

As specified in our theoretical study, it is possible to share a layer instance among several nodes (fig. 77). This is implemented as a factory, which produces mobility processes. The layers are used as a template from which a LayeredMobileProcess is created for each node of the group. The *last_shared_layer* attribute indicates which layers are shared. Individual layers are a copy of the instances provided to the template. Shared layers are realized with the help of queues, each of the nodes having a reference to the common queue. The nodes consume the data stored in the queue, and if any of the nodes reaches the tail of the queue, the corresponding layer is executed and its output is enqueued. When all nodes part of a given group have consumed part of the data, that part is released.

B.4 HYBRID MODELS

Creating hybrid models is a powerful feature of the simulator. It is possible to change node's current movement process by checking conditions specified by the scenario creator. The conditions are evaluated on certain events, such as:

- On a given time instant (e.g. every 100 s.)
- Before and after a change in the internal state of the current LEMMA process (e.g. after the strategy layer has selected the destination zone)

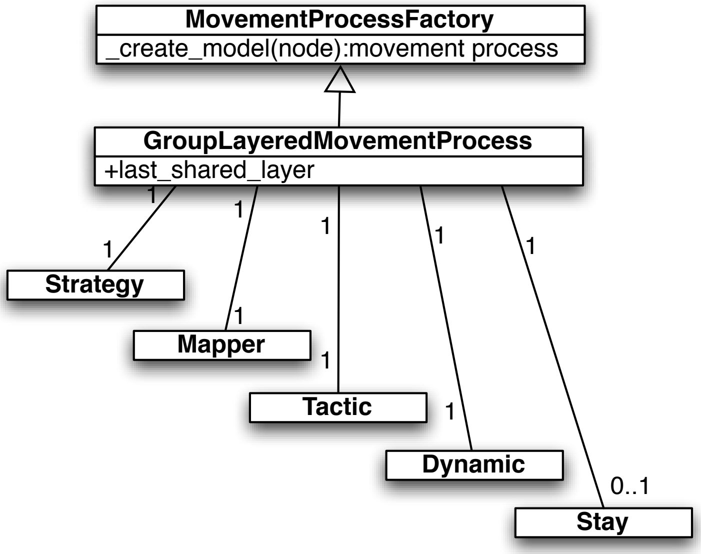


Figure 78: Representation of the objects of the group factory.

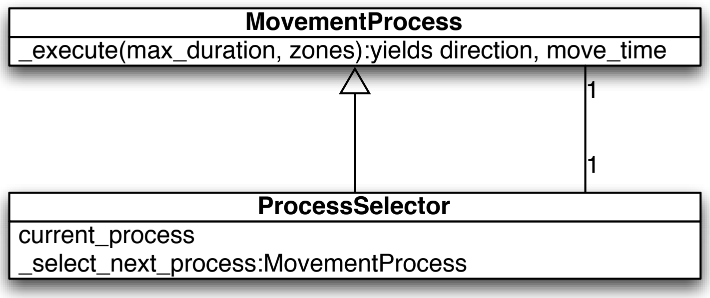


Figure 79: Process selector implementation diagram.

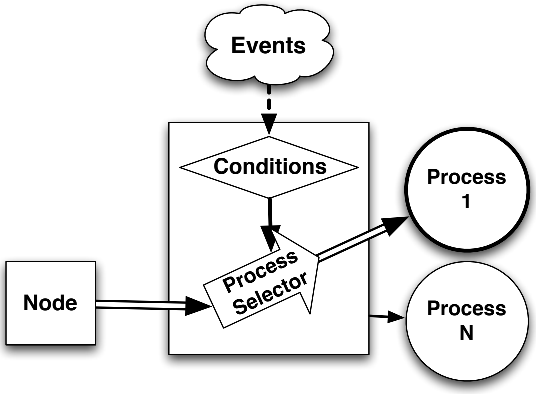


Figure 80: Multiple processes may be dynamically switched to create a hybrid node movement process.

- Triggered by the current movement process (e.g. the current process sends - "activate the next process")
- Upon a user-defined event (e.g. if a user clicks on a button in the GUI, or if a data packet is received)

Every time one of these event occurs, the conditions are evaluated, and a decision is made whether the process should be changed, or only a part of it (one or more layers). The case when the entire process is changed is demonstrated schematically in fig. 80. Selecting a different layer during the model execution is depicted in fig. 81. The internal state of a LEMMA process is reported before and after the execution of each of the layers. Thus, it is possible to create scenarios that alter the behavior of the node based on the decisions of the layers (e.g. if the strategy selects a distant zone, the dynamic may be changed, so that the node would move faster, thus simulating getting on a car). Furthermore, it is possible to act upon the arrival of user-defined events, such as the arrival of a data packet (see Section B.5). For example, if node A receives a packet from node B indicating that they should meet in a given zone, the process selector could switch the active process to one that is going to satisfy that request.

Each event provides specific information - for example the time instant event gives the current time, the event fired before a given layer is executed contains the layer input (e.g. a zone for the mapper), while the event fired after the layer execution contains its output (e.g. a point for the mapper). At that point, it is up to a function written by the scenario creator to evaluate the conditions he/she deems relevant (e.g. distance to the destination point) and act accordingly (do nothing, request the activation of another layer or process). Moreover, the scenario creator may chose to utilize some of the predefined condition evaluation functions.

B.5 COOPERATIVE SIMULATION WITH A NETWORK SIMULATOR

The mobility simulator outputs traces, which can then be fed into ns-2 or any other network simulator supporting traces. This method of offline trace generation however does not allow for the nodes to adjust their behavior based on the data transferred over the network. In order

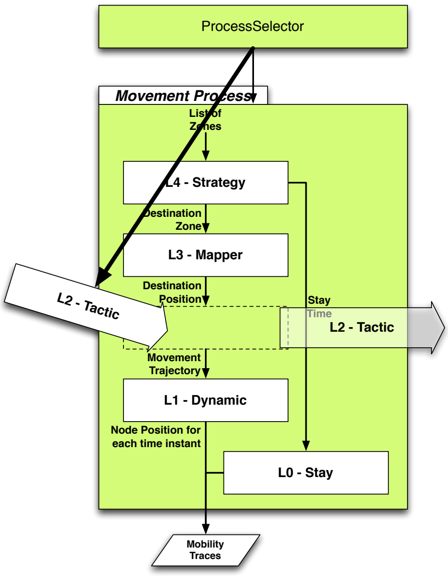


Figure 81: Fine-grained hybrid node behavior may be achieved by changing a single layer of its process.

to enable such interaction, we have implemented the possibility to invoke remote procedures using JSON-RPC - a remote procedure call protocol based on JSON [Cro06]. We chose JSON-RPC because it is a lightweight and simple, yet flexible and expressive protocol. Our implementation works over Netstrings [Ber] (a self-delimiting encoding of strings) over TCP or UDP.

The network simulator acts as a server, which executes the requests given by the mobility simulator. We have fixed a set of functions exposed via JSON-RPC by the network simulator. They allow the two simulators to synchronize their internal clocks (via the functions "get_current_time", "callback_at", "stop_at", "run", etc.), the mobility simulator to control the positions of the nodes in the network simulator (via "set_position", "set_direction", "set_acceleration"), send and receive data ("send", "receive"), etc. Thus, any network simulator providing these functions via JSON-RPC can perform a simulation cooperatively with LEMMA.

Activating this functionality is achieved with a single line of code. Once the LEMMA simulation is started, every change in any of the nodes is going to be pushed to the network simulator. The communicator object can be then used to send packets, and fires events every time a new packet is received.

The solution has been successfully implemented for the ns-3 network simulator [ns3].

B.6 USING LEMMA

In this section we are going to describe the ways LEMMA can be used to create various mobility models ranging from simple layered models, to complex hybrid or group models.

B.6.1 *Simple mobility models*

The Random Waypoint Mobility Model can be described with LEMMA by using the following code:

```
mp = LayeredMovementProcess(node, environment)
mp.strategy = UniformStrategy(min_pause, max_pause)
mp.mapper = random_mapper
mp.tactic = linear_tactics
mp.dynamic = ConstantMovementDynamic(1,5)
mp.stay = PauseStay()
```

The strategy is to select randomly the destination zone, then pick up randomly the destination point, move linearly to that point at a constant speed (taken in the range [1m/s, 5m/s]), and then pause during the stay in that zone.

In order to change the behavior of our model and add graph constraints to the movement pattern, one only has to change the tactic:

```
mp.tactic = GraphTactics()
```

If we want the node to deviate slightly from the linear trajectory specified by the graph, we can add some "oscillations" to the dynamic:

```
mp.dynamic = OscillatingMovementDynamic(
               ConstantMovementDynamic(1, 5))
```

With the same ease we can change any of the layers, including the stay layer. Here, we are using another LEMMA movement process for the stay layer:

```
mp.stay = LayeredMovementProcess(None, environment)
mp.stay.strategy = SameZoneStrategy()
mp.stay.mapper = border_mapper
mp.stay.tactic = linear_tactics
mp.stay.dynamic = ConstantMovementDynamic(1, 5)
mp.stay.stay = None
```

B.6.2 *Group mobility models*

Creating a group mobility pattern requires the creation of a movement process factory. If we want all nodes to have shared strategy, mapper and tactic, we can create the factory like this:

```
group = GroupLayeredMovementProcessFactory(
               last_shared_layer = Layers.tactic, environment)

group.strategy = UniformStrategy(min_pause, max_pause)
group.mapper = random_mapper
group.tactic = linear_tactics
group.dynamic = ConstantMovementDynamic(1, 5)
group.stay = PauseStay()
```

and then assign that behavior to several nodes:

```
for i in range(0, 100):
    node = Node(i)
    node.movement_process = group.create_process(node)
```

B.6.3 *Hybrid mobility models*

Implementing a hybrid model is achieved with the help of the ProcessSelector. It receives as a parameter a function, which based on its inputs, chooses the next node process, or modifies the current process itself. For example, defining a process, which moves with speed [20m/s,25m/s] when the destination zone is more than 3km away, and [1m/s, 5m/s] otherwise can be achieved with the following selector:

```
class DistanceLayerSelector(ProcessSelector):
    ...
    def _select_next_process(self,
                after_layer=None, state=None):
        area = self.environment.simulation_area
        dist = area.distance(self.node.position,
                    state['dest_zone'])
        if dist < self.threshold_distance:
            self.process.dynamic = self.dynamic1
        else:
            self.process.dynamic = self.dynamic2
```

After which, the only thing to do is to create the layered model, and activate the selector:

```
for i in range(0, 100):
    node = Node(i)

    process = LayeredMovementProcess()
    process.strategy = UniformStrategy()
    process.mapper = random_mapper
    process.tactic = linear_tactics
    process.dynamic = None
    process.stay = None

    dynamic1 = ConstantMovementDynamic(5, 10)
    dynamic2 = ConstantMovementDynamic(25, 30)

    node.movement_process =
               DistanceLayerSelector(
               node, environment, process,
               dynamic1, dynamic2, 3000)
```

B.7 RELATED WORK

There exist several frameworks that can be used to create mobility models. One of the first to appear was the BonnMotion [bon] - a Java-based, open-source simulator supporting a limited set of simple mobility models. Another pioneering framework is IMPORTANT [BSH], which also provides limited mobility modeling capabilities.

There are several advanced mobility modeling frameworks, such as SWANS [swa], CanuMobiSim [Ste]/VanetMobiSim [van] and SUMO [sum], as well as a myriad of individual simulation tools written for a single mobility model. Our architecture has several advantages when compared to these architectures, coming from the core principles of its theoretical foundation, namely: standardized node environment, each model is divided into five layers, each layer has distinct functions and layers communicate via simple interfaces.

The simplicity of these principles permits the easy creation of hybrid and group mobility models - a unique feature of our framework. Furthermore, creating new models is as easy as selecting the different layers and integrating them into new combinations. Adding a single new layer greatly increases the number of available models. Moreover, providing the possibility to perform cooperative simulations with an external network simulator exists only in SUMO, but its implementation is based on a proprietary protocol in contrast to our platform- and language- independent, standards-based approach. Our architecture is also supported by many studies performed in other scientific fields, such as traffic engineering, urban planning, evacuation studies, etc. [PN08].

B.8 conclusion

In this section we have presented *LEMMA*'s open-source, multiagent microsimulator. It can be used to construct a wide variety of models, including the majority of microscopic models used in wireless network simulations. The major entities of the simulation environment have been described and implemented, along with the way the different layers interact with them. We have also given the means of building hybrid and group models.

C LIST OF FIGURES AND TABLES

LIST OF FIGURES

LIST OF TABLES

[ABD+04] Alain Abran, Pierre Bourque, Robert Dupuis, James W. Moore, and Leonard L. Tripp, editors. *Guide to the Software Engineering Body of Knowledge - SWEBOK*. IEEE Press, Piscataway, NJ, USA, 2004 version edition, 2004.

[Abr98] T. Abrahamsson. Estimation of origin-destination matrices using traffic count–a literature survey, 1998.

[AFMT04] N. Aschenbruck, M. Frank, P. Martini, and J. Tolle. Human mobility in manet disaster area simulation - a realistic approach. In *Local Computer Networks, 2004. 29th Annual IEEE International Conference on*, pages 668–675, 2004.

[AG92] Kay W. Axhausen and Tommy Gärling. Activity-based approaches to travel analysis: conceptual frameworks, models, and research problems. *Transport Reviews*, 12(4):323 – 341, June 1992.

[AS03] Daniel Ashbrook and Thad Starner. Using gps to learn significant locations and predict movement across multiple users. *Personal Ubiquitous Comput.*, 7(5):275–286, 2003.

[AS07a] D. N. Alparslan and K. Sohraby. A generalized random mobility model for wireless ad hoc networks and its analysis: One-dimensional case. *Networking, IEEE/ACM Transactions on*, 15(3):602–615, 2007.

[AS07b] D. N. Alparslan and K. Sohraby. Two-dimensional modeling and analysis of generalized random mobility models for wireless ad hoc networks. *Networking, IEEE/ACM Transactions on*, 15(3):616–629, 2007.

[Ass89] Tranplan Associates. Waterloo region travel survey 1987: An overview of the survey findings, October 1989.

[AUS01] Sungjoon Ahn and A. Udaya Shankar. Adapting to route-demand and mobility (arm) in ad hoc network routing. *Network Protocols, 2001. Ninth International Conference on*, pages 44–52, 2001.

[AYS+09] A. Ahmed, K. Yasumoto, N. Shibata, T. Kitani, and M. Ito. Dar: Distributed adaptive service replication for manets. In *Wireless And Mobile Computing, Networking And Communications, 2009. (WiMob'2009), IEEE International Conference on*, 2009.

[B+96] M. Bergamo et al. System design specification for mobile multimedia wireless network mmwn. Technical Report DARPA Project DAAB07-95-C-D156, DARPA, October 1996.

[BC03] Magdalena Balazinska and Paul Castro. Characterizing mobility and network usage in a corporate wireless local-area network. In *MobiSys '03: Proceedings of the 1st international conference on Mobile systems, applications and services*, pages 303–316, New York, NY, USA, 2003. ACM.

[BCSW98] Stefano Basagni, Imrich Chlamtac, Violet R. Syrotiuk, and Barry A. Woodward. A distance routing effect algorithm for mobility (dream). In *MobiCom '98: Proceedings of the 4th annual ACM/IEEE international conference on Mobile computing and networking*, pages 76–84, New York, NY, USA, 1998. ACM Press.

[BD99] Amiya Bhattacharya and Sajal K. Das. Lezi-update: an information-theoretic approach to track mobile users in pcs networks. In *MobiCom '99: Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking*, pages 1–12, New York, NY, USA, 1999. ACM.

[BDdAF05] V. Borrel, M. Dias de Amorim, and S. Fdida. A preferential attachment gathering mobility model. *Communications Letters, IEEE*, 9(10):900–902, 2005.

[BDGH93] A. Banerjee, J. J. Dolado, J. W. Galbraith, and D. F. Hendry. *Cointegration, Error Correction, and the Econometric Analysis of Non-Stationary Data*. Oxford University Press, Oxford, 1993.

[Ber] D. J. Bernstein. Netstrings. `http://cr.yp.to/proto/netstrings.txt`.

[Bet01a] Christian Bettstetter. Mobility modeling in wireless networks: categorization, smooth movement, and border effects. *SIGMOBILE Mob. Comput. Commun. Rev.*, 5(3):55–66, 2001.

[Bet01b] Christian Bettstetter. Smooth is better than sharp: a random mobility model for simulation of wireless networks. In *MSWIM '01: Proceedings of the 4th ACM international workshop on Modeling, analysis and simulation of wireless and mobile systems*, pages 19–27, New York, NY, USA, 2001. ACM Press.

[BGB01] Ljubica Blazevic, Silvia Giordano, and Jean-Yves Le Boudec. Self organized terminode routing simulation. In *MSWIM '01: Proceedings of the 4th ACM international workshop on Modeling, analysis and simulation of wireless and mobile systems*, pages 81–88, New York, NY, USA, 2001. ACM Press.

[BH04] Fan Bai and Ahmed Helmy. *Wireless Ad Hoc and Sensor Networks*, chapter A Survey Of Mobility Models in Wireless Adhoc Networks. Kluwer Academic Publishers, June 2004.

[BHK05] David P. Blinn, Tristan Henderson, and David Kotz. Analysis of a wi-fi hotspot network. In *WitMeMo '05: International Workshop on Wireless Traffic Measurements and Modeling*, 2005.

[BHM07] Rainer Baumann, Simon Heimlicher, and Martin May. Towards realistic mobility models for vehicular ad-hoc networks. In *IEEE INFOCOM 2007, MOVE Workshop, 26th Annual IEEE Conference on Computer Communications*, May 2007.

[BKOKR04] T. Breyer, M. Klein, P. Obreiter, and B. König-Ries. Activity-based user modeling in service-oriented ad-hoc-networks. In *First Working Conference on Wireless On-Demand Network Systems (WONS 2004)*, 2004.

[BL04] Ken Blakely and Bruce Lowekamp. A structured group mobility model for the simulation of mobile ad hoc networks. In *MobiWac '04: Proceedings of the second international workshop on Mobility management & wireless access protocols*, pages 111–118, New York, NY, USA, 2004. ACM Press.

[BLdAF06] Vincent Borrel, Franck Legendre, Marcelo Dias de Amorim, and Serge Fdida. Simps: Using sociology for personal mobility. *CoRR*, abs/cs/0612045, 2006.

[BLDAF09] Vincent Borrel, Franck Legendre, Marcelo Dias De Amorim, and Serge Fdida. Simps: using sociology for personal mobility. *IEEE/ACM Trans. Netw.*, 17(3):831–842, 2009.

[BNKS95] Amotz Bar-Noy, Ilan Kessler, and Moshe Sidi. Mobile users: to update or not to update? *Wirel. Netw.*, 1(2):175–185, 1995.

[bon] Bonnmotion. `http://web.informatik.uni-bonn.de/IV/Mitarbeiter/dewaal/BonnMotion/`.

[Bra99] Plamen Bratanov. *User Mobility Modeling in Cellular Communications Networks*. PhD thesis, Technischen Universität Wien - Fakultät für Elektrotechnik, 1999.

[BRC+04] D. Bhattacharjee, A. Rao, Shah C, M. Shah, and A. Helmy. Empirical modeling of campus-wide pedestrian mobility observations on the usc campus. In *Vehicular Technology Conference, 2004. VTC2004-Fall. 2004 IEEE 60th*, volume 4, pages 2887–2891 Vol. 4, 2004.

[BRS02] Douglas M. Blough, Giovanni Resta, and Paolo Santi. A statistical analysis of the long-run node spatial distribution in mobile ad hoc networks. In *MSWiM '02: Proceedings of the 5th ACM international workshop on Modeling analysis and simulation of wireless and mobile systems*, pages 30–37, New York, NY, USA, 2002. ACM.

[BRS03] Christian Bettstetter, Giovanni Resta, and Paolo Santi. The node distribution of the random waypoint mobility model for wireless ad hoc networks. *IEEE Transactions on Mobile Computing*, 2(3):257–269, 2003.

[BRS04] Sven Bittner, Wolf-Ulrich Raffel, and Manuel Scholz. Heterogeneous mobility models and their impact on data dissemination in mobile ad-hoc networks. Technical report, N/A, Aug. 2004.

[BRS05] S. Bittner, W. U. Raffel, and M. Scholz. The area graph-based mobility model and its impact on data dissemination. In *Pervasive Computing and Communications Workshops, 2005. PerCom 2005 Workshops. Third IEEE International Conference on*, pages 268–272, 2005.

[BSH] F. Bai, Narayanan Sadagopan, and A. Helmy. Important. `http://nile.cise.ufl.edu/important/`.

[BSH03] F. Bai, Narayanan 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 Societies. IEEE*, volume 2, pages 825–835, 2003.

[BV04] J. Le Boudec and M. Vojnovi. Perfect simulation and stationarity of a class of mobility models. Technical report, 2004.

[cab] Cabspotting. `http://cabspotting.org`.

[Car96] Kathleen M. Carley. Validating computational models. Technical report, Carnegie Mellon University, September 1996.

[CB05] J. Capka and R. Boutaba. A mobility management tool-the realistic mobility model. In *Wireless And Mobile Computing, Networking And Communications, 2005. (WiMob'2005), IEEE International Conference on*, volume 2, pages 242–246 Vol. 2, 2005.

[CBD02] T. Camp, J. Boleng, and V. Davies. 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(5):483–502, 2002.

[CE07] Han Cai and Do Young Eun. Crossing over the bounded domain: from exponential to power-law inter-meeting time in manet. In *MobiCom '07: Proceedings of the 13th annual ACM international conference on Mobile computing and networking*, pages 159–170, New York, NY, USA, 2007. ACM.

[Cen] U.S. Census. Tiger. `http://www.census.gov/geo/www/tiger/`.

[CHC+05] Augustin Chaintreau, Pan Hui, Jon Crowcroft, Christophe Diot, Richard Gass, and James Scott. Pocket switched networks, or human mobility patterns as part of store-and-forward, or story-and-carry data transmission. Technical Report UCAM-CL-TR-617, University of Cambridge Computer Laboratory, February 2005.

[CHC+06] A. Chaintreau, P. Hui, J. Crowcroft, C. Diot, R. Gass, and J. Scott. Impact of human mobility on the design of opportunistic forwarding algorithms. In *INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings*, pages 1–13, 2006.

[CHD+07] Augustin Chaintreau, Pan Hui, Christophe Diot, Richard Gass, and James Scott. Impact of human mobility on opportunistic forwarding algorithms. *IEEE Transactions on Mobile Computing*, 6(6):606–620, 2007. Fellow-Crowcroft, Jon.

[CLF07] Vania Conan, Jérémie Leguay, and Timur Friedman. Characterizing pairwise inter-contact patterns in delay tolerant networks. In *Autonomics '07: Proceedings of the 1st international conference on Autonomic computing and communication systems*, pages 1–9, ICST, Brussels, Belgium, Belgium, 2007. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).

[CMS03] Fred Choa, Ronald T Milam, and David Stanek. Corsim, paramics, and vissim: What the manuals never told you. In *Proceedings of the Ninth TRB Conference on the Application of Transportation Planning Methods, April 6-10, 2003, Baton Rouge, Louisiana*, 2003.

[COdM04] C. A. V. Campos, D. C. Otero, and L. F. M. de Moraes. Realistic individual mobility markovian models for mobile ad hoc networks. In *Wireless Communications and Networking Conference, 2004. WCNC. 2004 IEEE*, volume 4, pages 1980–1985 Vol.4, 2004.

[cra] Crawdad. `http://www.crawdad.org/`.

[Cro06] Douglas Crockford. The application/json media type for javascript object notation (json). RFC 4627, Internet Engineering Task Force, July 2006.

[CSN07] Aaron Clauset, Cosma Rohilla Shalizi, and M. E. J. Newman. Power-law distributions in empirical data, 2007.

[Dav00] Vanessa Ann Davies. Evaluating mobility models within an ad hoc network. Master's thesis, Colorado School of Mines, 2000.

[DPR00] Samir Ranjan Das, Charles E. Perkins, and Elizabeth E. Royer. Performance comparison of two on-demand routing protocols for ad hoc networks. In *INFOCOM (1)*, pages 3–12, 2000.

[drm] Japan digital road map. `http://www.drm.jp/english/drm/e_index.htm`.

[ea79] Schlesinger et al. Terminology for model credibility. *Simulation*, 32(3):103–104, 1979.

[Ein05] Albert Einstein. Zur theorie der brownschen bewegung. *Annalen der Physik*, 17:549, 1905.

[EL04] S.M. Eisenman and G.F. List. Using probe data to estimate od matrices. *Intelligent Transportation Systems, 2004. Proceedings. The 7th International IEEE Conference on*, pages 291–296, Oct. 2004.

[EP09] Nathan Eagle and Alex Pentland. Eigenbehaviors: identifying structure in routine. *Behavioral Ecology and Sociobiology*, pages 1057–1066, May 2009.

[Fel94] Martin Fellendorf. VISSIM: A microscopic simulation tool to evaluate actuated signal control including bus priority. In *Proceedings of the 64th ITE Annual Meeting*, Dallas, TX, USA, 1994. Institute of Traffic Engineers.

[FERH06] Hector Flores, Stephan Eidenbenz, Rudolf Riedi, and Nick Hengartner. Describing manets: principal component analysis of sparse mobility traces. In *PE-WASUN '06: Proceedings of the 3rd ACM international workshop on Performance evaluation of wireless ad hoc, sensor and ubiquitous networks*, pages 123–131, New York, NY, USA, 2006. ACM.

[FHFB07] Marco Fiore, Jerome Harri, Fethi Filali, and Christian Bonnet. Vehicular mobility simulation for vanets. In *ANSS '07: Proceedings of the 40th Annual Simulation Symposium*, pages 301–309, Washington, DC, USA, 2007. IEEE Computer Society.

[GBNQ06] Joy Ghosh, Matthew J. Beal, Hung Q. Ngo, and Chunming Qiao. On profiling mobility and predicting locations of wireless users. In *REALMAN '06: Proceedings of the second international workshop on Multi-hop ad hoc networks: from theory to reality*, pages 55–62, New York, NY, USA, 2006. ACM Press.

[GHB] Marta C. Gonzalez, Cesar A. Hidalgo, and Albert-Laszlo Barabasi. Understanding individual human mobility patterns - supplimentary information. `http://www.nature.com/nature/journal/v453/n7196/full/nature06958.html`.

[GHB08] Marta C. Gonzalez, Cesar A. Hidalgo, and Albert-Laszlo Barabasi. Understanding individual human mobility patterns. *Nature*, 453(7196):779–782, June 2008.

[Gol95] Reginald Golledge. Path selection and route preference in human navigation: A progress report. In *Spatial Information Theory A Theoretical Basis for GIS*, volume 988/1995 of *Lecture Notes in Computer Science*, pages 207–222. Springer Berlin / Heidelberg, 1995.

[Gol99] Reginald G. Golledge, editor. *Wayfinding Behavior, Cognitive Mapping and Other Spatial Processes*. The Johns Hopkins University Press, Baltimore & London, 1999.

[GPQ05] J. Ghosh, S. J. Philip, and C. Qiao. Sociological orbit aware location approximation and routing in manet. In *Broadband Networks, 2005 2nd International Conference on*, pages 641–650 Vol. 1, 2005.

[GSN05] B. Gloss, M. Scharf, and D. Neubauer. A more realistic random direction mobility model, COST 290 TD(05)052. In *Proceedings of the 4th COST 290 Management Committee Meeting; Würzburg, Germany, Oct. 13-14, 2005*, pages 0–0. N/A, October 2005.

[GSN06] B. Gloss, M. Scharf, and D. Neubauer. Location-dependent parameterization of a random direction mobility model. In *Vehicular Technology Conference, 2006. VTC 2006-Spring. IEEE 63rd*, volume 3, pages 1068–1072, May 2006.

[Gue87] R. A. Guerin. Channel occupancy time distribution in a cellular radio system. *Vehicular Technology, IEEE Transactions on*, 36(3):89–99, 1987.

[HFB06] Jérôme Haerri, Fethi Filali, and Christian Bonnet. Mobility models for vehicular ad hoc networks: a survey and taxonomy. Technical report, Institut Eurecom, France, Mar 2006.

[HGPC99] Xiaoyan Hong, Mario Gerla, Guangyu Pei, and Ching-Chuan Chiang. A group mobility model for ad hoc wireless networks. In *MSWiM '99: Proceedings of the 2nd ACM international workshop on Modeling, analysis and simulation of wireless and mobile systems*, pages 53–60, New York, NY, USA, 1999. ACM Press.

[HH05] W.-j. Hsu and A. Helmy. IMPACT: Investigation of Mobile-user Patterns Across University Campuses using WLAN Trace Analysis. *ArXiv Computer Science e-prints*, aug 2005.

[HH06a] W. Hsu and A. Helmy. Analyzing principal characteristics of user association patterns and eigen-behavior in wireless lan traces, 2006.

[HH06b] Wei-jen Hsu and A. Helmy. On modeling user associations in wireless lan traces on university campuses. In *Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks, 2006 4th International Symposium on*, pages 1–9, 2006.

[HJ00] Yih-Chun Hu and David B. Johnson. Caching strategies in on-demand routing protocols for wireless ad hoc networks. In *MobiCom '00: Proceedings of the 6th annual international conference on Mobile computing and networking*, pages 231–242, New York, NY, USA, 2000. ACM Press.

[HKA04] Tristan Henderson, David Kotz, and Ilya Abyzov. The changing usage of a mature campus-wide wireless network. In *In Proceedings of ACM MOBICOM*, pages 187–201. ACM Press, 2004.

[HKA08] Tristan Henderson, David Kotz, and Ilya Abyzov. The changing usage of a mature campus-wide wireless network. *Comput. Netw.*, 52(14):2690–2712, 2008.

[HKG+01] Xiaoyan Hong, Taek Jin Kwon, Mario Gerla, Daniel Lihui Gu, and Guangyu Pei. A mobility framework for ad hoc wireless networks. *Lecture Notes in Computer Science*, 1987:185–??, 2001.

[HLL+07] Hong-Yu Huang, Pei-En Luo, Minglu Li, Da Li, Xu Li, Wei Shu, and Min-You Wu. Performance evaluation of suvnet with real-time traffic data. *IEEE Transactions on Vehicular Technology*, 56(6, Part 1):3381–3396, Nov. 2007.

[HLV06] E. Hyytia, P. Lassila, and J. Virtamo. Spatial node distribution of the random waypoint mobility model with applications. *Mobile Computing, IEEE Transactions on*, 5(6):680–694, June 2006.

[HMS+04] Wei-jen Hsu, K. Merchant, Haw-wei Shu, Chih-hsin Hsu, and A. Helmy. Preference-based mobility model and the case for congestion relief in wlans using ad hoc networks. In *Vehicular Technology Conference, 2004. VTC2004-Fall. 2004 IEEE 60th*, volume 4, pages 2962–2966 Vol. 4, 2004.

[HMS+05] Wei-jen Hsu, Kashyap Merchant, Haw-wei Shu, Chih-hsin Hsu, and Ahmed Helmy. Weighted waypoint mobility model and its impact on ad hoc networks. *SIGMOBILE Mob. Comput. Commun. Rev.*, 9(1):59–63, 2005.

[HMT08] L. Hanzo, S.M. Mostafavi, and R. Tafazolli. Connectivity-related properties of mobile nodes obeying the random walk and random waypoint mobility models. In *Vehicular Technology Conference, 2008. VTC Spring 2008. IEEE*, pages 133–137, May 2008.

[HMZ05] Christian Roman Hans-Martin Zimmermann, Ingo Gruber. A voronoi-based mobility model for urban environments. *European Wireless (EW'05)*, Apr. 2005.

[Hoo04] Serge Hoogendoorn. Pedestrian flow modeling by adaptive control. *Transportation Research Record: Journal of the Transportation Research Board*, 1878:95–103, 2004.

[HR86] Daehyoung Hong and S. S. Rappaport. Traffic model and performance analysis for cellular mobile radio telephone systems with prioritized and nonprioritized handoff procedures. *Vehicular Technology, IEEE Transactions on*, 35(3):77–92, 1986.

[Hua05] Dijiang Huang. Using delaunay triangulation to construct obstacle detour mobility model. In *Wireless Communications and Networking Conference, 2005 IEEE*, volume 3, pages 1644–1649 Vol. 3, 2005.

[HZ02] R. Hutchins and E.W. Zegura. Measurements from a campus wireless network. In *Communications, 2002. ICC 2002. IEEE International Conference on*, volume 5, pages 3161–3167 vol.5, 2002.

[HZW+07] Jun Hong, Xuedan Zhang, Zhongya Wei, Li Li, and Yong Ren. Spatial and temporal analysis of probe vehicle-based sampling for real-time traffic information system. In *Intelligent Vehicles Symposium, 2007 IEEE*, pages 1234–1239, June 2007.

[IR05] M. U. Ilyas and H. Radha. The influence mobility model: a novel hierarchical mobility modeling framework. In *Wireless Communications and Networking Conference, 2005 IEEE*, volume 3, pages 1638–1643 Vol. 3, 2005.

[JBC02] W. Navidi J. Boleng and T. Camp. Metrics to enable adaptive protocols for mobile ad hoc networks. In *Proceedings of the International Conference on Wireless Networks (ICWN'02)*, pages 293–298, 2002.

[JBRAS03] Amit Jardosh, Elizabeth M. Belding-Royer, Kevin C. Almeroth, and Subhash Suri. Towards realistic mobility models for mobile ad hoc networks. In *MobiCom '03: Proceedings of the 9th annual international conference on Mobile computing and networking*, pages 217–229, New York, NY, USA, 2003. ACM Press.

[JBRAS05] A. P. Jardosh, E. M. Belding-Royer, K. C. Almeroth, and S. Suri. Real-world environment models for mobile network evaluation. *Selected Areas in Communications, IEEE Journal on*, 23(3):622–632, 2005.

[JGQ04] Sumesh J. Philip Joy Ghosh and Chunming Qiao. Orbit mobility framework and orbit based routing (obr) protocol for manet. Technical report, CSE Department, University at Buffalo, Aug. 2004.

[jHDH07] Wei jen Hsu, Debojyoti Dutta, and Ahmed Helmy. Mining behavioral groups in large wireless lans. In *MobiCom '07: Proceedings of the 13th annual ACM international conference on Mobile computing and networking*, pages 338–341, New York, NY, USA, 2007. ACM.

[JLH+99a] Per Johansson, Tony Larsson, Nicklas Hedman, Bartosz Mielczarek, and Mikael Degermark. Scenario-based performance analysis of routing protocols for mobile ad-hoc networks. In *MobiCom '99: Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking*, pages 195–206, New York, NY, USA, 1999. ACM Press.

[JLH+99b] Per Johansson, Tony Larsson, Nicklas Hedman, Bartosz Mielczarek, and Mikael Degermark. Scenario-based performance analysis of routing protocols for mobile ad-hoc networks. In *MobiCom '99: Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking*, New York, NY, USA, 1999. ACM Press.

[JM96] David B. Johnson and David A. Maltz. Dynamic source routing in ad hoc wireless networks. In Imielinski and Korth, editors, *Mobile Computing*, volume 353. Kluwer Academic Publishers, 1996.

[JWZ01] Rui Jiang, Qingsong Wu, and Zuojin Zhu. Full velocity difference model for a car-following theory. *Physical Review E*, 64(1), 2001.

[Kal02] Olav Kallenberg. *Foundations of Modern Probability*. Springer, 2nd edition, 2002.

[KB05] Jonghyun Kim and Stephan Bohacek. A survey based mobility model of people for urban mesh networks. *MeshNets'05*, 2005.

[KCC05] Stuart Kurkowski, Tracy Camp, and Michael Colagrosso. Manet simulation studies: the incredibles. *SIGMOBILE Mob. Comput. Commun. Rev.*, 9(4):50–61, 2005.

[KE02] David Kotz and Kobby Essien. Analysis of a campus-wide wireless network. In *In Proceedings of ACM Mobicom*, pages 107–118. ACM Press, 2002.

[KK05] Minkyong Kim and David Kotz. Modeling users' mobility among wifi access points. In *WiTMeMo '05: Papers presented at the 2005 workshop on Wireless traffic measurements and modeling*, pages 19–24, Berkeley, CA, USA, 2005. USENIX Association.

[KKK06] Minkyong Kim, David Kotz, and Songkuk Kim. Extracting a mobility model from real user traces. In *Proceedings of the 25th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, Barcelona, Spain, April 2006. IEEE Computer Society Press.

[KLBV07] Thomas Karagiannis, Jean-Yves Le Boudec, and Milan Vojnović. Power law and exponential decay of inter contact times between mobile devices. In *MobiCom '07: Proceedings of the 13th annual ACM international conference on Mobile computing and networking*, pages 183–194, New York, NY, USA, 2007. ACM.

[Kle99] Jack P. C. Kleijnen. Validation of models: statistical techniques and data availability. In *WSC '99: Proceedings of the 31st conference on Winter simulation*, pages 647–654, New York, NY, USA, 1999. ACM Press.

[KML07] F. K. Karnadi, Zhi H. Mo, and Kun-Chan Lan. Rapid generation of realistic mobility models for vanet. In *Wireless Communications and Networking Conference, 2007.WCNC 2007. IEEE*, pages 2506–2511, 2007.

[KMR05] A. Khelil, P. J. Marron, and K. Rothermel. Contact-based mobility metrics for delay-tolerant ad hoc networking. *Modeling, Analysis, and Simulation of Computer and Telecommunication Systems, 2005. 13th IEEE International Symposium on*, pages 435–444, 2005.

[KRKB06] Birgitta König-Ries, Michael Klein, and Tobias Breyer. Activity-based user modeling in wireless networks. *Mob. Netw. Appl.*, 11(2):267–277, 2006.

[KSMK05] Herbert Kluepfel, M. Schreckenberg, and T. Meyer-Koenig. *Traffic and Granular Flow '03*, chapter Models for Crowd Movement and Egress Simulation, pages 357 – 372. 2005.

[Kul05] E. D. Kuligowski. Review of 28 egress models. In MD Gaithersburg, R. D. Peacock, and E. D. Kuligowski, editors, *Workshop on Building Occupant Movement During Fire Emergencies. Proceedings. Session 4.4. June 10-11, 2004*, pages 68–90. NIST SP 1032, January 2005.

[KYYM03] S. Kitamura, T. Yamamoto, T. Yoshii, and T. Miwa. Descriptive analysis of detouring traffic though neighborhood streets using probe car data. In *Proceedings of the 10th World Congress on Intelligent Transport Systems*, November 2003.

[LBdAF06a] Franck Legendre, Vincent Borrel, Marcelo Dias de Amorim, and Serge Fdida. Modeling mobility with behavioral rules: The case of incident and emergency situations. In Kenjiro Cho and Philippe Jacquet, editors, *AINTEC*, volume 4311 of *Lecture Notes in Computer Science*, pages 186–205. Springer, 2006.

[LBdAF06b] Franck Legendre, Vincent Borrel, Marcelo Dias de Amorim, and Serge Fdida. Reconsidering microscopic mobility modeling for self-organizing networks. *Network, IEEE*, 20(6):4–12, 2006.

[LBV05a] Jean-Yves Le Boudec and M. Vojnovic. Perfect simulation and stationarity of a class of mobility models. In *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, volume 4, pages 2743–2754 vol. 4, 2005.

[LBV05b] Jean-Yves Le Boudec and Milan Vojnovic. The Random Trip Model, Part II: Stationary Regime and perfect Simulation. Technical report, Microsoft Research - Microsoft Corporation, 2005. This is the extended version of the Infocom 2005 paper "Perfect Simulation and Stationarity of a Class of Mobility Models".

[LBV06a] J. Y. Le Boudec and M. Vojnovic. The random trip model: Stability, stationary regime, and perfect simulation. *Networking, IEEE/ACM Transactions on*, 14(6):1153–1166, 2006.

[LBV06b] Jean-Yves Le Boudec and Milan Vojnovic. The Random Trip Model: Stability, Stationary Regime, and Perfect Simulation. Technical report, Microsoft Research - Microsoft Corporation, 2006.

[LC02] J.R. Linton and Catherine. A comparison of selective initialization bias elimination methods. In *Simulation Conference, 2002. Proceedings of the Winter*, volume 2, pages 1951–1957 vol.2, Dec. 2002.

[LCJ05] J. Lee, S. Choi, and H. Jung. Analysis of user behavior and traffic pattern in a large-scale 802.11a/b network. In *Workshop on Wireless Network Measurements*, 2005.

[LH98] T. Larsson and N. Hedman. Routing protocols in wireless ad hoc networks - a simulation study. Master's thesis, Lulea Tekniska Universitet, 1998.

[LHK04] Pasi Lassila, Esa Hyytiä, and Henri Koskinen. Connectivity properties of random waypoint mobility model for ad hoc networks. In *in Proc. MedHoc-Net, Île de Porquerolles*, pages 159–168, 2004.

[LHK+08] Kyunghan Lee, Seongik Hong, Seong Joon Kim, Injong Rhee, and Song Chong. Demystifying levy walk patterns in human walks. Technical report, School of EECS, KAIST, Daejeon, Korea and Dept. of Computer Science, NC State University, Raleigh, NC. USA, 2008.

[LHK+09] Kyunghan Lee, Seongik Hong, Seong J. Kim, Injong Rhee, and Song Chong. Slaw: A mobility model for human walks. In *Proceedings of the 28th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, Rio de Janeiro, Brazil, April 2009. IEEE.

[LLN06] Peter Laborczi, Martin Linauer, and Bernhard Nowotny. Travel time estimation based on incomplete probe car information. In *13th World Congress on ITS*, 2006.

[LTYY04] Hsien-Chou Liao, Yi-Wei Ting, Shih-Hsuan Yen, and Chou-Chen Yang. Ant mobility model platform for network simulator. In *Information Technology: Coding and Computing, 2004. Proceedings. ITCC 2004. International Conference on*, volume 2, pages 380–384 Vol.2, 2004.

[LWM03] I. Lubashevsky, P. Wagner, and R. Mahnke. Bounded rational driver models. *The European Physical Journal B - Condensed Matter and Complex Systems*, 32(2):243–247, 2003.

[LYD06] Sunho Lim, Chansu Yu, and Chita R. Das. Clustered mobility model for scale-free wireless networks. In *Local Computer Networks, Proceedings 2006 31st IEEE Conference on*, pages 231–238, 2006.

[Mar97] J.G. Markoulidakis. Mobility modeling in third generation mobile telecommunication systems. *IEEE PCS*, Aug. 1997.

[Marar] Robert Marks. Validating simulation models: A general framework and four applied examples. *Computational Economics*, 2008, to appear.

[McK05] Erin McKean, editor. *The New Oxford American Dictionary*. Number ISBN 0-19-517077-6. Oxford University Press, second edition edition, May 2005.

[MHMA06] A. R. Momen, A. S. Hassani, A. Mirzaee, and P. Azmi. Intelligent vehicle mobility modeling based on a sub-optimal path finding method. In *Vehicular Technology Conference, 2006. VTC 2006-Spring. IEEE 63rd*, volume 6, pages 3012–3015, 2006.

[MM03] Tomio Miwa and Takayuki Morikawa. Analysis on route choice behavior based on probe-car data. In *Proceedings of the 10th World Congress on ITS*, 2003.

[MM06] Mirco Musolesi and Cecilia Mascolo. A community based mobility model for ad hoc network research. In *REALMAN '06: Proceedings of the second international workshop on Multi-hop ad hoc networks: from theory to reality*, pages 31–38, New York, NY, USA, 2006. ACM Press.

[MMLR05] Daniel Minder, Pedro José Marrón, Andreas Lachenmann, and Kurt Rothermel. Experimental construction of a meeting model for smart office environments. In *Proceedings of the First Workshop on Real-World Wireless Sensor Networks (REALWSN 2005), SICS Technical Report T2005:09*, June 2005.

[MPGW05] Atulya Mahajan, Niranjan Potnis, Kartik Gopalan, and An-I A. Wang. Evaluation of mobility models for vehicular ad-hoc network simulations. Technical report, Dept. of Computer Science, Florida State University, 2005.

[MSK+05] Kumiko Maeda, Kazuki Sato, Kazuki Konishi, Akiko Yamasaki, Akira Uchiyama, Hirozumi Yamaguchi, Keiichi Yasumoto, and Teruo Higashino. Getting urban pedestrian flow from simple observation: realistic mobility generation in wireless network simulation. In *MSWiM '05: Proceedings of the 8th ACM international symposium on Modeling, analysis and simulation of wireless and mobile systems*, pages 151–158, New York, NY, USA, 2005. ACM Press.

[MSM04] Tomio Miwa, Takaaki Sakai, and Taka Morikawa. Route identification and travel time prediction using probe-car data. *International Journal of ITS Research*, 2(1):21–28, October 2004.

[MTF04] Kantaro Monobe, Shigenori Tanaka, and Hitoshi Furata. Development of a system for making guide maps based on the idea of the cognitive map. In *International Conference on Computing in Civil and Building Engineering , ICCCBE , 10 , 2004.06.02-04 , Weimar , Bauhaus-Universität*. Professur Informatik im Bauwesen, 2004.

[MV05] Marvin McNett and Geoffrey M. Voelker. Access and mobility of wireless pda users. *SIGMOBILE Mob. Comput. Commun. Rev.*, 9(2):40–55, 2005.

[MWR+06] Rahul Mangharam, Daniel Weller, Raj Rajkumar, Priyantha Mudalige, and Fan Bai. Groovenet: A hybrid simulator for vehicle-to-vehicle networks. In *Mobile and Ubiquitous Systems: Networking & Services, 2006 Third Annual International Conference on*, pages 1–8, 2006.

[Nav04] William Navidi. Stationary distributions for the random waypoint mobility model. *IEEE Transactions on Mobile Computing*, 3(1):99–108, 2004. Member-Camp, Tracy.

[NBG06] Valery Naumov, Rainer Baumann, and Thomas Gross. An evaluation of inter-vehicle ad hoc networks based on realistic vehicular traces. In *MobiHoc '06: Proceedings of the 7th ACM international symposium on Mobile ad hoc networking and computing*, pages 108–119, New York, NY, USA, 2006. ACM.

[net] Networkx. `http://networkx.lanl.gov/`.

[NHH+06] T. Nakajima, S. Hashimoto, K. Haruyama, T. Nakamura, and Y. Osana. Office layout support system using interactive genetic algorithm. *Evolutionary Computation, 2006. CEC 2006. IEEE Congress on*, pages 56–63, 0-0 2006.

[NKK02] S. Nousiainen, K. Kordybach, and P. Kemppi. User distribution and mobility model framework for cellular network simulations. In *Proc. IST Mobile and Wireless Telecommunications Summit*, pages 518–522. unknown, May 2002.

[NNB+99] Kai Nagel, Kai Nagel, Richard J. Beckman, Richard J. Beckman, Christopher L. Barrett, and Christopher L. Barrett. Transims for urban planning, 1999.

[ns3] Ns3. `http://www.nsnam.org/`.

[otICS99] Software Engineering Technical Committee of the IEEE Computer Society, editor. *IEEE Standards: Software Engineering*, volume Volume Two: Process Standards, chapter IEEE Standard for Software Unit Testing: An American National Standard, ANSI/IEEE Std 1008-1987. The Institute of Electrical and Electronics Engineers, Inc., 1999.

[PCBH03] Xavier Pérez-Costa, Christian Bettstetter, and Hannes Hartenstein. Toward a mobility metric for comparable & reproducible results in ad hoc networks research. *SIGMOBILE Mob. Comput. Commun. Rev.*, 7(4):58–60, 2003.

[PDN07] Alexander Pelov, Pierre David, and Thomas Noel. 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*, pages 1–6, April 2007.

[PM00] Dan Pelleg and Andrew W. Moore. X-means: Extending k-means with efficient estimation of the number of clusters. In *Seventeenth International Conference on Machine Learning*, pages 727–734. Morgan Kaufmann, 2000.

[PM06] Niranjan Potnis and Atulya Mahajan. Mobility models for vehicular ad hoc network simulations. In *ACM-SE 44: Proceedings of the 44th annual Southeast regional conference*, pages 746–747, New York, NY, USA, 2006. ACM Press.

[PMK06] M. Papadopouli, M. Moudatsos, and M. Karaliopoulos. Modeling roaming in large-scale wireless networks using real measurements. In *World of Wireless, Mobile and Multimedia Networks, 2006. WoWMoM 2006. International Symposium on a*, page 6 pp., 2006.

[PN08] Alexander Pelov and Thomas Noel. Layered architecture for mobility models - lemma. In *3rd International Conference on Broadband Communications, Information Technology and Biomedical Applications (BroadCom' 08)*, 2008.

[PRL+08] M. Piórkowski, M. Raya, A. Lezama Lugo, P. Papadimitratos, M. Grossglauser, and J.-P. Hubaux. Trans: realistic joint traffic and network simulator for vanets. *SIGMOBILE Mob. Comput. Commun. Rev.*, 12(1):31–33, 2008.

[PSDG08] Michal Piorkowski, Natasa Sarafijanovic-Djukic, and Matthias Grossglauser. On Clustering Phenomenon in Mobile Partitioned Networks. In *The First ACM SIGMOBILE International Workshop on Mobility Models for Networking Research*, pages 1–8. ACM, 2008.

[PSDG09a] Michal Piorkowski, Natasa Sarafijanovic-Djukic, and Matthias Grossglauser. CRAWDAD data set epfl/mobility (v. 2009-02-24). Downloaded from http://crawdad.cs.dartmouth.edu/epfl/mobility, February 2009.

[PSDG09b] Michal Piorkowski, Natasa Sarafijanovoc-Djukic, and Matthias Grossglauser. A Parsimonious Model of Mobile Partitioned Networks with Clustering. In *THE First International Conference on COMmunication Systems and NETworkS (COMSNETS)*, pages 1–10, 2009.

[PSS05a] M. Papadopouli, H. Shen, and M. Spanakis. Modeling client arrivals at access points in wireless campus-wide networks. In *Local and Metropolitan Area Networks, 2005. LANMAN 2005. The 14th IEEE Workshop on*, page 6 pp., 2005.

[PSS05b] Maria Papadopouli, Haipeng Shen, and Manolis Spanakis. Characterizing the duration and association patterns of wireless access in a campus. In *11th European Wireless Conference*, Nicosia, Cyprus, 2005.

[Ray03] Suprio Ray. Realistic mobility for manet simulation. Master's thesis, The University of British Columbia, December 2003.

[RBA05] Andres Rojas, Philip Branch, and Grenville Armitage. Experimental validation of the random waypoint mobility model through a real world mobility trace for large geographical areas. In *MSWiM '05: Proceedings of the 8th ACM international symposium on Modeling, analysis and simulation of wireless and mobile systems*, pages 174–177, New York, NY, USA, 2005. ACM Press.

[RMSM01] E.M. Royer, P.M. Melliar-Smith, and L.E. Moser. An analysis of the optimum node density for ad hoc mobile networks. In *Communications, 2001. ICC 2001. IEEE International Conference on*, volume 3, pages 857–861 vol.3, 2001.

[ROPT05] M. Raento, A. Oulasvirta, R. Petit, and H. Toivonen. Contextphone: a prototyping platform for context-aware mobile applications. *Pervasive Computing, IEEE*, 4(2):51–59, April 2005.

[RSH+08] Injong Rhee, Minsu Shin, Seongik Hong, Kyunghan Lee, and Song Chong. On the levy-walk nature of human mobility. In *Proceedings of the 27th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, Arizona, USA, April 2008. IEEE.

[San] Sanchez. Nomadic, pursue and column mobility models. `http://www.disca.upv.es/misan/mobmodel.htm`.

[Sar98] Robert G. Sargent. Verification and validation of simulation models. In *WSC '98: Proceedings of the 30th conference on Winter simulation*, pages 121–130, Los Alamitos, CA, USA, 1998. IEEE Computer Society Press.

[SB04] D. Schwab and R. Bunt. Characterising the use of a campus wireless network. In *INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies*, volume 2, pages 862–870 vol.2, March 2004.

[SBKH03] Narayanan Sadagopan, Fan Bai, Bhaskar Krishnamachari, and Ahmed Helmy. Paths: analysis of path duration statistics and their impact on reactive manet routing protocols. In *MobiHoc '03: Proceedings of the 4th ACM international symposium on Mobile ad hoc networking & computing*, pages 245–256, New York, NY, USA, 2003. ACM Press.

[Sch06] Christian Schindelhauer. Mobility in wireless networks. In *32nd International Conference on Current Trends in Theory and Practice of Computer Science (SOFSEM 2006)*, Merin, Czech Republic, 21 - 27 2006.

[SCJ04] K. D. Simler, S. E. Czerwinski, and A. D. Joseph. Analysis of wide area user mobility patterns. In *Mobile Computing Systems and Applications, 2004. WMCSA 2004. Sixth IEEE Workshop on*, pages 30–40, 2004.

[Sco97] John Scourias. Dynamic location management and activity-based mobility modelling for cellular networks. Master of mathematics in computer science, University of Waterloo, Ontario, Canada, 1997.

[SCP+04] Jing Su, A. Chin, A. Popivanova, A. Goel, and E. de Lara. User mobility for opportunistic ad-hoc networking. In *Mobile Computing Systems and Applications, 2004. WMCSA 2004. Sixth IEEE Workshop on*, pages 41–50, 2004.

[SD84] SAID E. SAID and DAVID A. DICKEY. Testing for unit roots in autoregressive-moving average models of unknown order. *Biometrika*, 71(3):599–607, 1984.

[SGS+06] Timothy Sohn, William G. Griswold, James Scott, Anthony LaMarca, Yatin Chawathe, Ian Smith, and Mike Chen. Experiences with place lab: an open source toolkit for location-aware computing. In *ICSE '06: Proceedings of the 28th international conference on Software engineering*, pages 462–471, New York, NY, USA, 2006. ACM.

[SHB+03] I. Stepanov, J. Hahner, C. Becker, Jing Tian, and K. Rothermel. A meta-model and framework for user mobility in mobile networks. In *Networks, 2003. ICON2003. The 11th IEEE International Conference on*, pages 231–238, 2003.

[SHH02] S. Shah, E. A. Hernandez, and A. S. Helal. Cad-hoc: a cad-like tool for generating mobility benchmarks in ad-hoc networks. In *Applications and the Internet, 2002. (SAINT 2002). Proceedings. 2002 Symposium on*, pages 270–279, 2002.

[SI01] Deepanshu Shukla and Sridhar Iyer. Mobility models in ad hoc networks, Nov. 2001.

[sim] Simpy. `http://simpy.sourceforge.net/`.

[Sio05] I. Siomina. A novel simulation approach for mobility models in an inhomogeneous area. In *Personal, Indoor and Mobile Radio Communications, 2005. PIMRC 2005. IEEE 16th International Symposium on*, volume 4, pages 2116–2120 Vol. 4, 2005.

[SJ04] Amit Kumar Saha and David B. Johnson. Modeling mobility for vehicular ad-hoc networks. In *Vehicular Ad Hoc Networks*, pages 91–92, 2004.

[SK] Cosma Rohilla Shalizi and Aryeh Kontorovich. Almost none of the theory of stochastic processes.

[SK99a] J. Scourias and T. Kunz. Activity-based mobility modeling: realistic evaluation of location management schemes for cellular networks. *Wireless Communications and Networking Conference, 1999. WCNC. 1999 IEEE*, 1(300), 1999.

[SK99b] John Scourias and Thomas Kunz. An activity-based mobility model and location management simulation framework. In *MSWiM '99: Proceedings of the 2nd ACM international workshop on Modeling, analysis and simulation of wireless and mobile systems*, pages 61–68, New York, NY, USA, 1999. ACM Press.

[SKJH04] Libo Song, David Kotz, Ravi Jain, and Xiaoning He. Evaluating location predictors with extensive Wi-Fi mobility data. In *Proceedings of the 23rd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, volume 2, pages 1414–1424, March 2004.

[SMO06] Vikram Srinivasan, Mehul Motani, and Wei Tsang Ooi. Analysis and implications of student contact patterns derived from campus schedules. In *MobiCom '06: Proceedings of the 12th annual international conference on Mobile computing and networking*, pages 86–97, New York, NY, USA, 2006. ACM.

[SMR05] I. Stepanov, P. J. Marron, and K. Rothermel. Mobility modeling of outdoor scenarios for manets. In *Simulation Symposium, 2005. Proceedings. 38th Annual*, pages 312–322, 2005.

[Son08] Libo Song. Evaluating mobility predictors in wireless networks for improving handoff and opportunistic routing. Technical Report TR2008-611, Dartmouth College, Computer Science, Hanover, NH, January 2008.

[SPH] Neil Schemenauer, Tim Peters, and Magnus Lie Hetland. Simple generators. `http://www.python.org/dev/peps/pep-0255/`.

[spo] Sportstracker. `http://sportstracker.nokia.com/`.

[Ste] Illya Stepanov. Canumobisim. http://canu.informatik.uni-stuttgart.de/mobisim/.

[Ste02] Illya Stepanov. *Integrating Realistic Mobility Models In Mobile Ad Hoc Network Simulation*. PhD thesis, Universität Stuttgart Fakultät Informatik, July 2002.

[sum] "simulation of urban mobility" (sumo). http://apps.sourceforge.net/mediawiki/sumo/.

[swa] Jist/swans. http://jist.ece.cornell.edu/.

[TBH05] Roy Timo, Kim Blackmore, and Leif Hanlen. On entropy measures for dynamic network topologies: Limits to manet. In *Proceedings of the Australian Communications Theory Workshop, AusCTW05*, pages 89 – 94, February 2005.

[TG05] C. Tuduce and T. Gross. A mobility model based on wlan traces and its validation. In *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, volume 1, pages 664–674 vol. 1, 2005.

[THB+02] Jing Tian, J. Hahner, C. Becker, I. Stepanov, and K. Rothermel. Graph-based mobility model for mobile ad hoc network simulation. In *Simulation Symposium, 2002. Proceedings. 35th Annual*, pages 337–344, 2002.

[THH00] Martin Treiber, Ansgar Hennecke, and Dirk Helbing. Congested traffic states in empirical observations and microscopic simulations. *Physical Review E*, 62:1805, 2000.

[THM+04] R. Takami, J. Hashiji, T. Miwa, T. Yamamoto, and T. Morikawa. Empirical analysis on multipurpose applicability of probe-car data. In *Proceedings of the 11th World Congress on Intelligent Transport Systems*, October 2004.

[TKH06] Martin Treiber, Arne Kesting, and Dirk Helbing. Delays, inaccuracies and anticipation in microscopic traffic models. *PHYSICA A*, 360:71, 2006.

[Tol99] V. Tolety. Load reduction in ad hoc networks using mobile servers. Master's thesis, Colorado School of Mines, 1999.

[TP06] Suttipong Thajchayapong and Jon M. Peha. Mobility patterns in microcellular wireless networks. *IEEE Transactions on Mobile Computing*, 5(1):52–63, 2006.

[TZH+02] Desney S Tan, Shuheng Zhou, Jiann-Min Ho, Janak S Mehta, and Hideaki Tanabe. Design and evaluation of an individually simulated mobility model in wireless ad hoc networks. In *Communication Networks and Distributed Systems Modeling and Simulation Conference 2002*, San Antonio, TX, 2002.

[van] Vanetmobisim. http://vanet.eurecom.fr/.

[Var06] A. Varshavsky. Are gsm phones the solution for localization? In *Mobile Computing Systems and Applications, 2006. WMCSA '06. Proceedings. 7th IEEE Workshop on*, pages 34–42, 2006.

[VF99] J. Voigt and G. P. Fettweis. Influence of user mobility and simulcast-handoff on the system capacity in pico-cellular environments. In *Wireless Communications and Networking Conference, 1999. WCNC. 1999 IEEE*, pages 712–716 vol.2, 1999.

[Vuo89] Quang H. Vuong. Likelihood ratio tests for model selection and non-nested hypotheses. *Econometrica*, 57(2):307–333, 1989.

[vZB82] Henk J. van Zuylen and David M. Branston. Consistent link flow estimation from counts. *Transportation Research Part B: Methodological*, 16(6):473–476, December 1982.

[Wal08] Brenton Walker. Using persistent homology to recover spatial information from encounter traces. In *MobiHoc '08: Proceedings of the 9th ACM international symposium on Mobile ad hoc networking and computing*, pages 371–380, New York, NY, USA, 2008. ACM.

[Wat03] Duncan J. Watts. *Small Worlds: The Dynamics of Networks between Order and Randomness (Princeton Studies in Complexity)*. Princeton University Press, illustrated edition edition, November 2003.

[WC09] S.Y. Wang and C.L. Chou. Nctuns tool for wireless vehicular communication network researches. *Simulation Modelling Practice and Theory*, 17(7):1211 – 1226, 2009.

[Wik08] Wikipedia. Duck typing — wikipedia, the free encyclopedia, 2008. [Online; accessed 30-November-2008].

[Wik09a] Wikipedia. Internet access — wikipedia, the free encyclopedia. `http://en.wikipedia.org/w/index.php?title=Internet_access&oldid=321613052`, 2009.

[Wik09b] Wikipedia. List of mobile network operators of europe — wikipedia, the free encyclopedia. `http://en.wikipedia.org/w/index.php?title=List_of_mobile_network_operators_of_Europe&oldid=311052291`, 2009. [Online; accessed 31-August-2009].

[WL02] K. H. Wang and Baochun Li. Group mobility and partition prediction in wireless ad-hoc networks. In *Communications, 2002. ICC 2002. IEEE International Conference on*, volume 2, pages 1017–1021 vol.2, 2002.

[WYMM06] L. Wang, T. Yamamoto, T. Miwa, and T. Morikawa. An analysis of effects of rainfall on travel speed at signalized surface road network based on probe vehicle data. In *Proceedings of ICTTS 2006*, pages 615–624, 2006.

[XBJ07] Sanlin Xu, Kim L. Blackmore, and Haley M. Jones. An analysis framework for mobility metrics in mobile ad hoc networks. *EURASIP J. Wirel. Commun. Netw.*, 2007(1):26–26, 2007.

[YLN03a] J. Yoon, M. Liu, and B. Noble. Random waypoint considered harmful. In *Proceedings of the 22nd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, volume 2, pages 1312–1321, April 2003.

[YLN03b] Jungkeun Yoon, Mingyan Liu, and Brian Noble. Sound mobility models. In *MobiCom '03: Proceedings of the 9th annual international conference on Mobile computing and networking*, pages 205–216, New York, NY, USA, 2003. ACM Press.

[YNLK06] Jungkeun Yoon, Brian D. Noble, Mingyan Liu, and Minkyong Kim. Building realistic mobility models from coarse-grained traces. In *MobiSys '06: Proceedings of the 4th international conference on Mobile systems, applications and services*, pages 177–190, New York, NY, USA, 2006. ACM Press.

[ZD97] M. M. Zonoozi and P. Dassanayake. User mobility modeling and characterization of mobility patterns. *Selected Areas in Communications, IEEE Journal on*, 15(7):1239–1252, 1997.

[ZHL06] Qunwei Zheng, Xiaoyan Hong, and Jun Liu. An agenda based mobility model. In IEEE Computer Society, editor, *Proceedings of the 39th Annual Simulation Symposium (ANSS'06)*, 2006.

[ZHR04] Qunwei Zheng, Xiaoyan Hong, and Sibabrata Ray. Recent advances in mobility modeling for mobile ad hoc network research. In *ACM-SE 42: Proceedings of the 42nd annual Southeast regional conference*, pages 70–75, New York, NY, USA, 2004. ACM Press.

[ZXG04] Biao Zhou, Kaixin Xu, and M. Gerla. Group and swarm mobility models for ad hoc network scenarios using virtual tracks. In *Military Communications Conference, 2004. MILCOM 2004. IEEE*, volume 1, pages 289–294 Vol. 1, 2004.

[ZXZ05] Liang Zou, Jian-Min Xu, and Ling-Xiang Zhu. Arterial speed studies with taxi equipped with global positioning receivers as probe vehicle. *Wireless Communications, Networking and Mobile Computing, 2005. Proceedings. 2005 International Conference on*, 2:1343–1347, Sept. 2005.

D PUBLICATIONS

Some ideas and figures have appeared previously in the following publications:

INTERNATIONAL JOURNALS

Layered Mobility Model Architecture - LEMMA
Alexander Pelov and Thomas Noël
In African Journal of Information & Communication Technology, vol. 5. issue. 1, year 2009, issn 1449-2679

INTERNATIONAL CONFERENCES

Trace Analysis of a Wireless University Network with Authentication
Alexander Pelov, Pierre David and Thomas Noël
In Proceedings of the Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks and Workshops, 2007. WiOpt 2007. 5th International Symposium on, pp. 1–6, Limassol, Cyprus, April 2007.

Layered Architecture for Mobility Models – LEMMA
Alexander Pelov and Thomas Noël
In Proceedings of the Broadband Communications, Information Technology & Biomedical Applications, 2008 Third International Conference on, pp. 365–372, Pretoria, South Africa, November 2008.

Creating advanced mobility models with LEMMA.
Alexander Pelov and Thomas Noël
In Proceedings of the 12th Communications and Networking Simulation Symposium (San Diego, USA, March 2009). CNS '09. ACM.

Mathematical Foundations Of The Layered Mobility Model Architecture - LEMMA
Alexander Pelov and Thomas Noël
In Proceedings of the Networking and Communications, 2009. WIMOB '09. IEEE International Conference on Wireless and Mobile Computing, Marrakech, Morocco, October 2009.

BOOK CHAPTERS

Layered Architecture For Mobility Models - LEMMA
Alexander Pelov and Thomas Noël
In Planning and Optimisation of 3G & 4G Wireless Networks, J.I. Agbinya (ed.)