PEDECIBA Informática
Instituto de Computación – Facultad de Ingeniería
Universidad de la República
Montevideo, Uruguay

Tesis de Maestría
en Informática

Demos connectivity model (DCM)

Martín Giachino

2013

Martín Giachino
Demos connectivity model (DCM)
ISSN 0797-6410
Tesis de Maestría en Informática
Reporte Técnico RT 13-02
PEDECIBA
Instituto de Computación – Facultad de Ingeniería
Universidad de la República.
Montevideo, Uruguay, 2013

Autor: Martín Giachino, giachino@fing.edu.uy

Título: Demos Connectivity Model

Tutor: Dr. Ing. Eduardo Grampín Castro, Instituto de Computación - Facultad de Ingeniería
Universidad de la República Montevideo, Uruguay

Fecha: 23/11/2012

Resumen: La explosión en el uso de las redes inalámbricas y las tecnologías móviles introduce nuevos requerimientos para la investigación y desarrollo de productos, servicios y protocolos que trabajan con estas tecnologías. La posibilidad de que los usuarios puedan utilizar la tecnología mientras se están moviendo hace necesario entender de forma cabal la movilidad, con el fin de predecir cómo se comportarán las aplicaciones en un contexto real. El método más común es probar estos escenarios mediante el uso de simuladores de red, lo que hace posible probar escenarios complejos sin el alto costo de hacer un despliegue real. Los simuladores de red son capaces de simular los movimientos de entidades móviles (como usuarios y dispositivos), pero se necesita un modelo de movilidad para gobernar el patrón de movimientos de estos nodos móviles que los modelen las características del escenario bajo estudio. El presente trabajo realiza un estudio detallado del estado del arte en modelos de movilidad y toma como caso de estudio una red real del Plan Ceibal. Los datos reales fueron recolectados y procesados con herramientas desarrolladas ad-hoc para este trabajo, y un nuevo modelo ha sido desarrollado usando patrones de movilidad derivados de los datos de movilidad y conectividad reales.

Palabras clave: redes inalámbricas, modelos de movilidad, simuladores de red.

Contents

Chapter 1

Introduction

1.1 Introduction

Widespread adoption in the use of wireless networks and mobile technologies introduces new requirements to the research and development of products, services, and protocols that work on these technologies. The possibility that users could use the technology while they are moving makes it necessary to fully understand mobility in order to predict how the applications will behave in a real context. The most common way in which these scenarios can be tested is through the use of network simulators, making it possible to test complex scenarios without the expensive cost of doing a real deployment. Network simulators are able to simulate node movements (like users and devices) but they need a mobility model to govern nodes pattern of movements that model the characteristics of the scenario under study.

A mobility model is defined as a representation of real agents, focused on the description of motion, based on knowledge of how certain parameters (connectivity, localization, speed, acceleration or positions) change over time. Since real scenarios are generally very complex, the reality is simplified by design decisions that do not affect significantly the modeled reality, in order to make possible the development of software tools or to derive analytical equations. Definition and use of mobility models is extensive and not only in technological areas, but also in a wide range of applications like transport vehicles, logistics, movements of populations in urban environments, animals in their natural habitat, environmental impacts of waste from vehicles, mobile technologies and networks, among many others.

The present work is focused in the emerging context of *opportunistic networks* and *delay-tolerant networking (DTN)* and proposes a mobility model for a particular *DTN*, formed by a set of *One Laptop Per Child* laptops *[105](OLPC)* that belongs to the *Plan Ceibal Project* [107]. *DTNs* are those that, contrary to what happens in structured networks, are formed by chance, and as long as two nodes remain ad-hoc connection is brief and intermittent, but where the information is still plausible to travel from origin to destination. Uruguay has a countrywide deployment of the *OLPC* initiative, locally named *Plan Ceibal*, ruled by the government, seeking to decrease the digital gap in the country by giving to every scholar the opportunity to access information and technology regardless of their location and/or socioeconomic environment. Although children are instructed at school, they can use their laptops anywhere, for instance at home, therefore impacting families' lives. *OLPC* laptops, also called *XO*, are low cost laptops with limited hardware resources designed by the *OLPC* project with the mission of *"empower the world's poorest children*

Figure 1.1: XO laptop.

through education"[1]. From the official startup of the project in 2006 and up until now, around 570.000 laptops have been delivered to schoolchildren all over the country, and children carry their laptops from home to school and back every day, making it possible to create new applications to take advantage of computers ubiquity, mobility, and wireless capabilities, converting this laptops in a large DTN spread all along the country. In this scenario, the project *Domestic Environment Monitoring with Opportunistic Sensor* networks (*DEMOS*) *[133]*, tries to use *Plan Ceibal* laptops as an *opportunistic* and *delay tolerant network* to obtain, transport, aggregate and communicate environmental information from sensors disseminated near of children's neighborhoods. These sensors are installed in fixed locations, completely independent of the children's laptops, which are used to collect and forward the sensor information, using wireless opportunistic networking techniques, while the children move around in their daily life. Later at school, using the same techniques, the data will be transmitted to an environment monitoring station using the Internet. This monitoring station may be operated by governmental or non-governmental organizations or even an on-line facility for the open control by the same community that is object of the monitoring.

This kind of networking imposes certain features to mobility models, which should be modeled to obtain an accurate prediction in the evaluation of new applications and protocols. Present work provides a detailed study of the state of the art on mobility models and takes as a case of study a real *Plan Ceibal* network. Real data was collected and processed by tools developed for this work, and a new model was built using mobility patterns derived from real data, attempting to be possible the use of network simulators to simulate *Plan Ceibal* networks and to test the application system developed for *DEMOS* project in scenarios similar to the real environment.

[1]See OLPC Project mission in http://one.laptop.org/about/mission

1.2 About This Document

Chapter 2 presents a full descriptive and detailed State of Art on mobility models, showing its pros and cons. Chapter 3 describes the scenario of the present case of study, the data to be available from a real network deployment and makes a complete comparison of the Mobility Models against the desired properties. Chapter 4 resumes the process to get the real data (design, implementation and data analysis). Chapter 5 defines and describe the new model developed. Chapter 6 covers evaluation of the proposed model against real data, and finally, Chapter 7 presents conclusions and future work.

Chapter 2

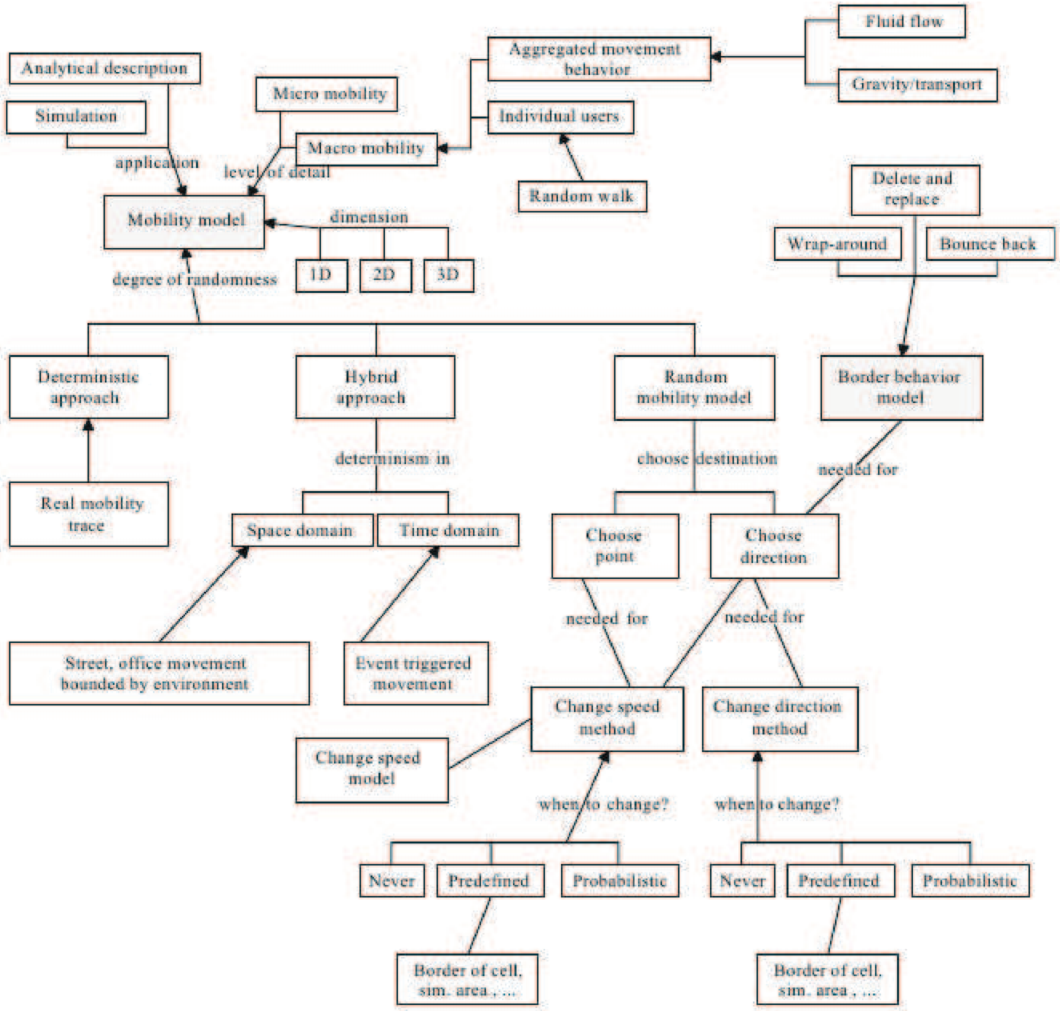
State of the Art

2.1 What is a mobility model?

There is no unique definition of what a mobility model is, but there are many partial definitions in the related work, making a descriptive definition, based on properties that a mobility model must reflect. For *Tuduce and Thomas [130]* a mobility model is a set of rules used to generate trajectories for mobile entities, while for *Camp et al [26]* a mobility model should attempt to mimic the movements of real mobile nodes (MNs). Changes in speed and direction must occur and they must occur in reasonable time slots. In Kim et al [66] is said: *"To develop a mobility model, we must understand user mobility. We must obtain detailed mobility data about real users, and carefully characterize their mobility. We analyzed mobility characteristics including pause time, speed, and direction of movements.....These mobility characteristics provide the fundamental information that underlies any mobility model"*. Capka and Boutaba [27] define a mobility model as *"a representation of a certain real or abstract world that contains moving entities. The world is said to exist for some finite amount of time during which each moving entity has one unique but changing location of presence as defined in the location granularity of the world"*. Finally, for Sichitiuin [121] a mobility model is defined as *"is a method of simulating movement of mobile nodes, usually for the purpose of further using the resulting movement for other simulations"*.

Gavrilovska and Prasadin [41] point that a mobility model design process should have in mind some general objectives: ***level of details***, ***dimension***, ***border behavior*** y ***degree of randomness*** (see Figure 2.1 on page 5). ***Level of details*** refers to granularity, and is divided in three groups: *microscopic* (describes movement of each node as its position and velocity in a certain time), *mesoscopic* (also called *kinetic*, describes the homogeneous motion of groups) and *macroscopic* (is interested in properties like density, average and deviation speed, etc). ***Dimension*** refers to the kind of motion (1D, 2D o 3D). **Border behavior** defines the decisions to take when nodes reach the edge of the modeled area. There are 3 possibilities: Bounce, Delete and replace, and Wrap (and Aschenbruck et al [4] added another option, the possibility that a node leaves the area). Finally, **degree of randomness** is the way nodes can move within the area, mainly, how are chosen the two most important parameters: direction and speed. Each node can move freely in certain ways, or even could have predefined paths.

Figure 2.1: Mobility models design concepts (taken from [41]).



2.2 Network simulation and mobility

The use of network simulators to evaluate performance of protocols and applications has grown enormously in recent years. Particularly in mobile networks, where it is essential to reflect upon the fact that nodes move during the simulation, it is imperative that these movements are fitted with good accuracy compared to reality, since the results of the simulations depend heavily on it [26, 62, 112]. That is why in recent years, the study and modeling of mobility has become a key area for the study of wireless networks, whether cellular, *WLAN*, *MANET* or *Ad-Hoc* networks.

In the context of network simulation, there are two ways to use mobility patterns of simulated entities, the use of traces and the use of synthetic models [26, 94]. Traces are measurements taken in already deployed networks, referred to data related to position of real entities (usually people and/or mobile devices) that are intended to simulate. On the other hand, synthetic models are abstractions of reality, shaped by mathematical equations that attempt to capture the main features of real mobile entities.

Based on this classification, both approaches shows advantages and disadvantages in its use, so depending on characteristics to model or evaluate, it could be more adequate the use of one or another.

Use of real traces (assumes the existence of an actual deployment, which is not applicable in many cases) is more accurate for particular scenario from where they were collected. As negative characteristics: first, it could be too expensive (or not possible) to get the data, and second, is not possible to change parameters on it, so there is no way to run simulations changing parameters to generate different scenarios.

On the other hand, use of *synthetic models* are much simple to generate and use, they have no cost to use because it is not necessary to previously get and process data. Furthermore, they present the ability to enable the modification of parameters, so simulations could explore a more complete set of probable scenarios. Even when these features are extremely interesting, the big problem resides in the fact that is not easy to reproduce real movement patterns in simple mathematic equations, so in many cases those models are far from reality.

Then, in the network simulation area, and particularly in the use of movement patterns, this is a brief summary of situation:

- Use of *synthetic models* is widely accepted and disseminated because they are easy to use. At the same time, the results should be analyzed carefully, having in mind that many of them are not reflecting characteristics of complex movement patterns good enough.
- Following the general problems mentioned above, *synthetic models* have evolved, trying to solve the undesirable properties which drive them away from reality.
- As a way to better reflect characteristics of real environments, the simple synthetic models evolve to the so called *synthetic hybrids models*, which combine two or more simple synthetic models.
- The direct use of real traces in the simulation of networks is not generally applicable, except where it is proposed to study a particular scenario, or when evaluating an existing model.
- Instead of using real traces directly, a previous process to get the fundamental properties of motion could be performed, and thus, attempt to create a more general model based on this data. In this way it aims to generate models that can vary parameters and make more general studies.
- Finally, new approaches are being used today to describe and define models. These models are not synthetic nor trace-based, but higher level strategies are met to model the node motion.

In this sense, a first possible generalization in two groups could be made [87]: *(1) Trace-to-model:* when real data is used to characterize motion patterns and to build a model with them, and *(2) Model-to-Trace* when, similar as the synthetic models, mathematical equations are used to describe movement of entities.

2.3 Classification

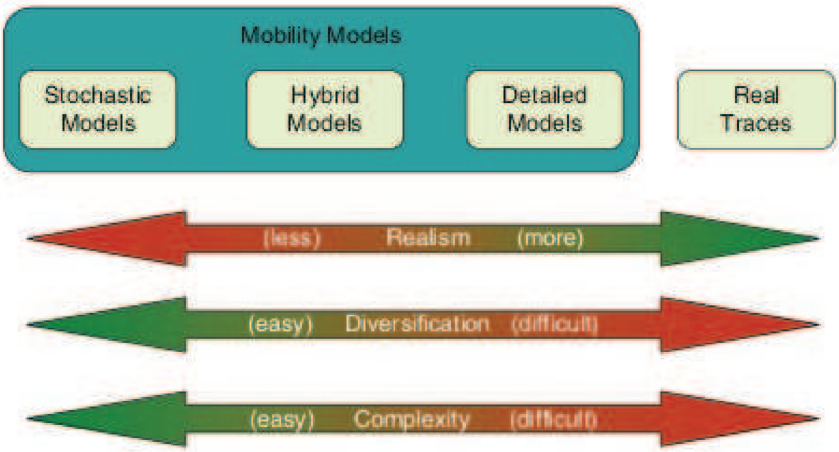
Many different model classifications exist in the literature, all of them with the objective of creating an order and grouping them by relevant properties, in order to easily understand advantages and disadvantages of each group of models.

2.3.1 Based on types and their advantages

The classification made by Sichitiu [121] takes three parameters for comparison: ***Realism***, ***Diversification*** and ***Complexity***. *Realism* refers to the level of accuracy achieved with respect to a real scenario, *Diversification* refers on how the model could fit to a different kind of scenarios (i.e. that permits different types of nodes like pedestrian and vehicles, or to different environments like cities, conferences or campus) and *complexity* intends to measure the level of traces computation needed for a simulation in the sense that a more complex model consumes more computational resources. Another possible classification could have four categories: *Stochastic*, *Detailed*, *Hybrid* and *Real traces*. Relationship with the parameters of comparison is shown in Figure 2.2 on page 7).

- ***Stochastic*** generates random movements without constraints.
- ***Detailed*** is a model built for a particular scenario.
- ***Hybrid*** is a mixture of the above models. Within these models there are group, obstacles and trace based mobility models.
- Last, ***Real traces***, are a set of trajectories of real users or nodes in a particular scenario. CRAWDAD is a repository of publicly available real traces [135].

Figure 2.2: Classification by pros and cons (taken from [121]).



Same authors still propose one more classification criterion, based on behavior and process of construction: ***Stochastic***, ***Groups***, ***Obstacle***, ***Detailed*** and ***Trace-Based*** mobility models.

- ***Stochastic:*** Similar to theoretical models because of their simplicity and not real characteristics.
- ***Groups:*** A little more complex than stochastic models, they try to reflect group behavior of people and animals.
- ***Obstacle:*** It includes presence of obstacles as a way to restrict trajectories of nodes.
- ***Detailed:*** Mode-specific scenarios, like vehicles on the street, conferences and campus among others.
- ***Trace-Based:*** It refers to the simulation starting from real traces, and to create new models based on the statistical information that characterizes the traces, allowing to vary parameters.

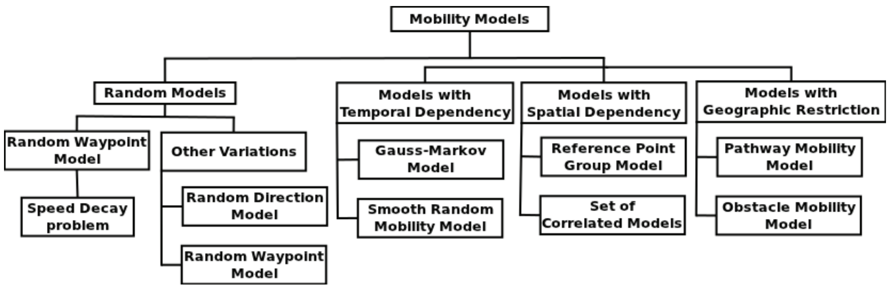
2.3.2 Based on movements restrictions and dependencies

Aschenbruck et al in [95] propose a more strict classification based on the intrinsic properties of the models:

- ***Random based:*** Without constraints nor dependencies.
- ***Temporal dependencies:*** Movements depends on the past movements.
- ***Spatial dependencies:*** Movements of a node depends on the nodes around.
- ***Geographic restrictions:*** Not all places in the area are allowed to be transited.
- ***Hybrid characteristics:*** When a model takes characteristics from two or more of the above models.

In [5] a brief schema is presented including some well known models and its categorization using the above classification as shown in Figure 2.3 on page 8.

Figure 2.3: Model classification (taken from [5])



2.3.3 Based on rules and original discipline

In the classification proposed in [41] models are divided in three groups: ***Traditional***, ***Enriched, and Interdisciplinary***. Traditional are those that model objects as entities or groups, leaving them to choose their speed and direction without constraints and without taking care of relation of the nodes around them. *Enriched* takes the same characteristics from [81] (who also defines *Contraction*, *Expansion* and *Circling* Mobility Models). Last category, *Interdisciplinary*, refers to the models generated from others theories like social theory [93], AI Game programming [23][75] among others.

2.4 Mobility models

The following is a summary of the best known mobility models, taking as guide the classification described in Section 2.3.2. This decision does not imply loss of generality, and could have been chosen any other classification. Which will be used was chosen because in this classification is more intuitive categorization of some models.

2.4.1 Random based mobility models

Models included in this category are also called as entity-models and have stochastic characteristics, because they choose the direction and speed in a random way regardless of previous movements, so they are memoryless models.

2.4.1.1 Random Walk mobility model

This is the simplest model, which was defined by Einstein, and it is also known under the name of *Brownian motion.* Attempts to represent the movement of many entities of nature, for example emulating the unpredictable movement of particles of physics.

The rules that govern the movement of each node are as follows:

- Each node changes its speed and direction at constant intervals of time t, or when a fixed distance d has been traveled (see Figure 2.4 on page 10 and Figure 2.5 on page 11 respectively).
- The new node direction $\theta(t)$ is obtained randomly and uniformly in the interval $(0, 2\pi]$
- And the new node velocity $v(t)$ is obtained randomly and uniformly in the interval $[0, V_{max}]$

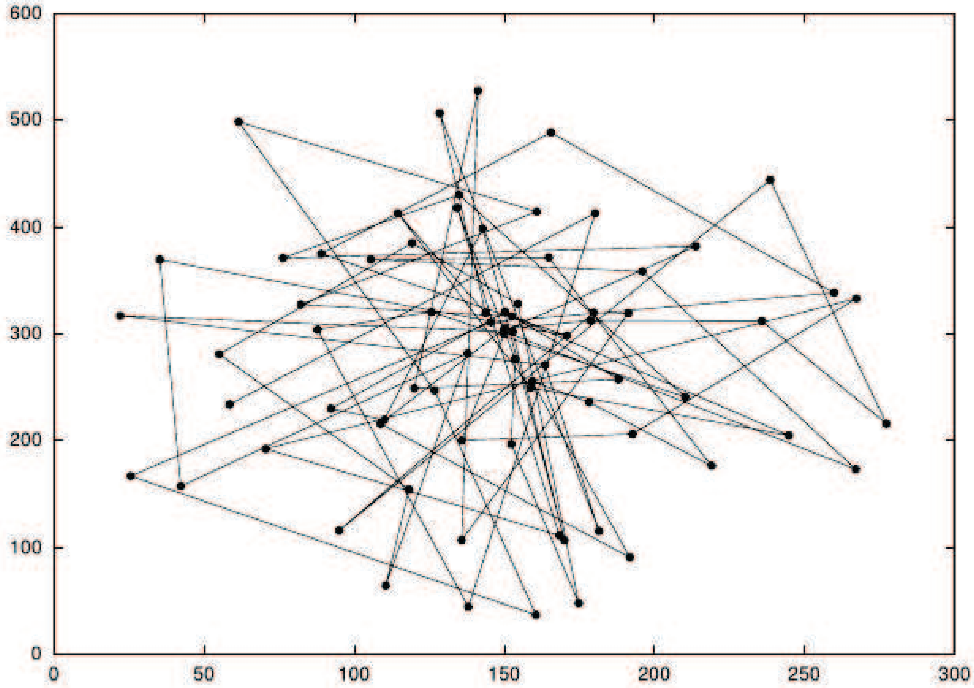
Others models derived from *Random Walk* are *1D*, *2D*, *3D,* and *dD walks* and *Random Waypoint Model, Markovian Waypoint Model (MWP)* or addition of attraction points [14]. More details could be found in *[26, 47].*

2.4.1.2 Random Waypoint mobility model

The *Random Waypoint Model (RWP)* [22] model is one of the most simple and widely used for mobile ad-hoc simulations. It describes movement patterns of individual nodes without any restrictions about their destination, direction and velocity. Movements of nodes are governed by this set of rules:

- Initially, nodes are spread on the simulation area uniformly.
- Nodes are in two possible states, paused and moving.
- In paused state, node remains in the same location for a certain time T_{pause}. Note that when $T_{pause} = 0$ it corresponds to the *Random Walk Model* (see 2.4.1.1).
- When T_{pause} is finished, node randomly select a location as its new destination, and velocity is selected in the same way from $[0, V_{max}]$ interval. After that, node begins to move to the new destination with the selected velocity.
- Once the node reached the desired location, the process begins again.

Figure 2.4: Motion pattern of a node using the Random Walk Mobility Model with constant time (taken from *[26]*).



Even when RWP is widely adopted, many issues had been seen about node distribution and others unwanted characteristics, next a summary of them are presented.

The initial position of nodes is important, because if the *average node neighbor* metric is analyzed, it presents high variability in the first seconds of the simulation. In [26] an example of this is shown (see Figure 2.6 on page 12) and three alternatives for this initialization are proposed: use final simulation positions from previous long run simulation, use an alternative initial node (not pure random) distribution, and discard an initial time period of each simulation. At the same time a complex relation between node speed and pause time is presented, based on the *link breakage* metric (Figure 2.7 on page 13), that shows a more stable network for some values of time and pause, and that pause times have more influence than velocity.

RWP has the effect that average velocity is decreasing over simulation time if $v_{min} = 0$ [136]. Another phenomenon is the so called *non-uniform spatial distribution*, in which nodes tends to concentrate in the middle of the simulation area (see Figure 2.8 on page 13) [12, 13, 14]. Another pathology is the *density wave* phenomenon, that makes nodes to periodically fluctuate its number of neighbors [31, 72, 79]. In [14] attraction points are defined (who have more probability to be chosen than any other point) in order to break with the *non-uniform spatial distribution*.

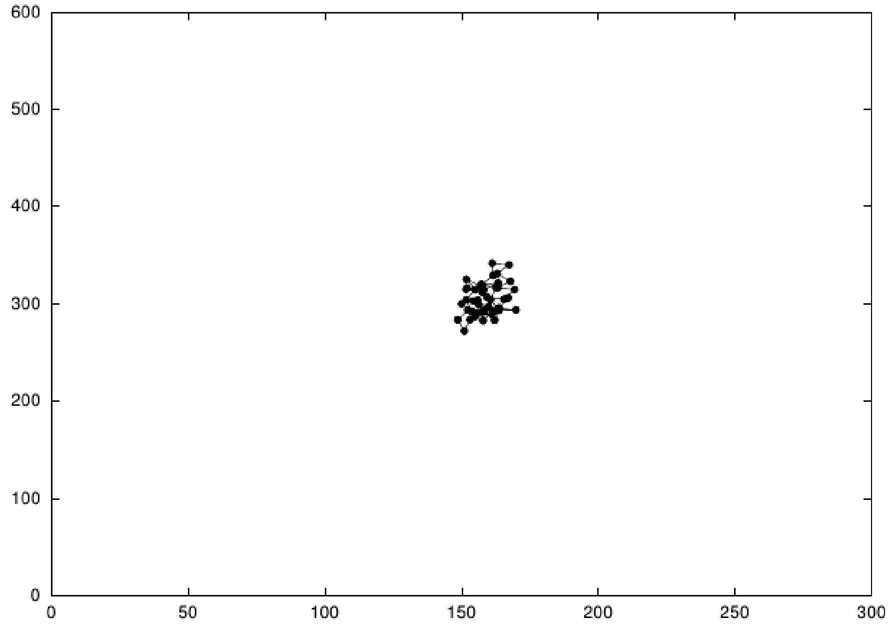


Figure 2.5: Motion pattern of a node using the Random Walk Mobility Model with constant distance (taken from *[26]*).

2.4.1.3 Random Direction mobility model

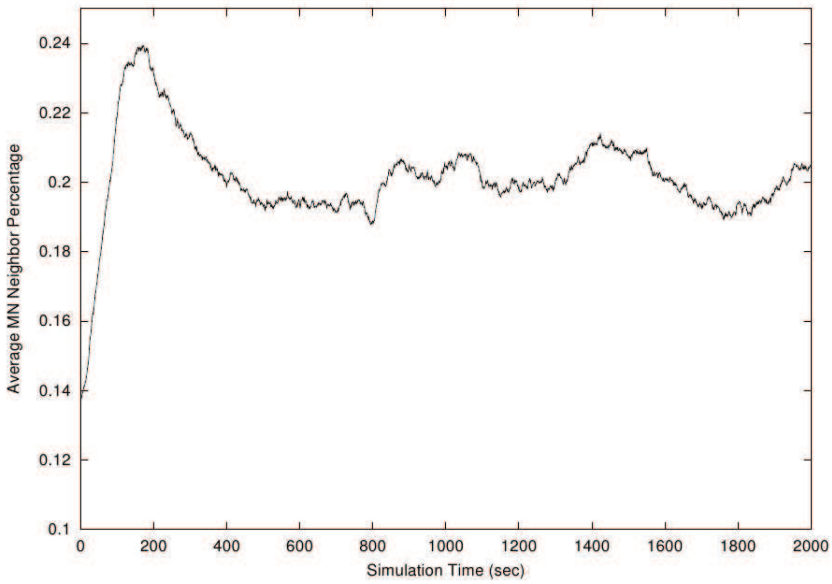
This model is defined in [119] as an alternative for the most used mobility model used in that moment, the *RWP* model. *Random Direction Mobility (RDM)* tries to solve the two main problems related to RWP, the *non-uniform spatial distribution* and *density wave* phenomenon, and movements of nodes are governed by this set of rules:

- At the beginning of the simulation, each node selects a degree in $[0, 2\pi]$, and then find a destination on the boundary in this direction of travel.
- It then selects a speed in $[v_{min}, v_{max}]$, and travels to that destination at the given speed
- Once it reaches the destination, it rests for the given pause time
- Then selects a new degree in $[0, \pi]$ (because the node is already on the boundary)
- The node then identifies the destination on the boundary in this line of direction, selects a new speed, and resumes travel

With this set of rules, nodes utilize all the simulation area and not concentrate in the middle. In figure 2.9 on page 14 (from [10]) an example of moving pattern for *RDM* is shown.

Authors of *RDM* also proposed a *"variation of the Random Direction model, called the Modified Random Direction model (MRDM). In this model, nodes select a direction degree as before, but they may choose their destination anywhere along that direction of travel. They do not need to travel all the way to the boundary"*.

Figure 2.6: Average neighbor percentage vs time (taken from *[26]*).



2.4.1.4 Random Borderpoint mobility model

Proposed in [14], it is a modification of *RWP* and, at the same time, it has similarities with *RDM*. The difference of this model with respect to *RWP* is that the position is chosen from the border area instead of from the whole area, and its main objective is to study node position distribution between them (in square and circular areas).

As a result of that work, authors conclude that node distribution of this new model in circular areas, not in square areas, presents a good uniform node distribution, as it could be seen in Figure 2.11 on page 16 and Figure 2.10 on page 15.

2.4.1.5 Clustered mobility model

Based on the fact that many social, natural, and biological networks are characterized by *scale-free power-law connectivity distribution* and a few densely populated nodes known as *hubs* [8, 125], authors in [78] proposed the *Clustered Mobility Model (CMM)*. The main goals of this model is to avoid temporal and spatial problems present in other random models, like the non-uniform distribution phenomenon, which makes nodes move closer to highly connected nodes, so clustered nodes around hubs are exhibited. In this way, *CMM* is designed to possess consistent steady-state mobility parameters.

CMM consists of two phases, *growth* and *rewiring* (based on the principle of preferential attachment). In the growth phase, the complete simulation area is logically divided in subareas. After that, nodes are located initially based on a probability value, to make node moves to choose with greater probability, a sub area highly populated than a sparse one, and once it has a sub area, randomly selects a position on it (see Equation 2.1).

Figure 2.7: Link breakage vs. speed vs. pause time (taken from *[26]*).

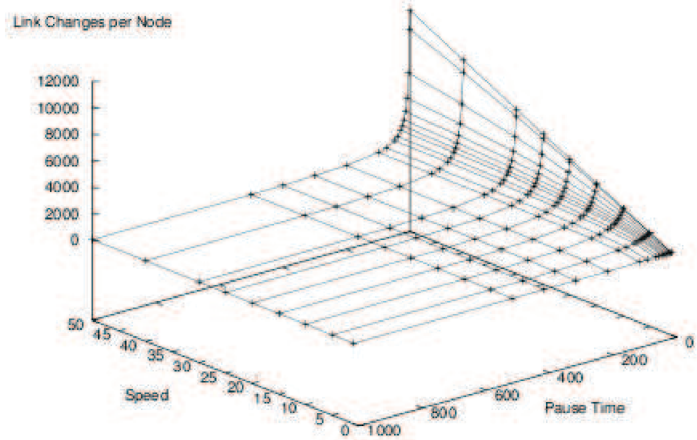
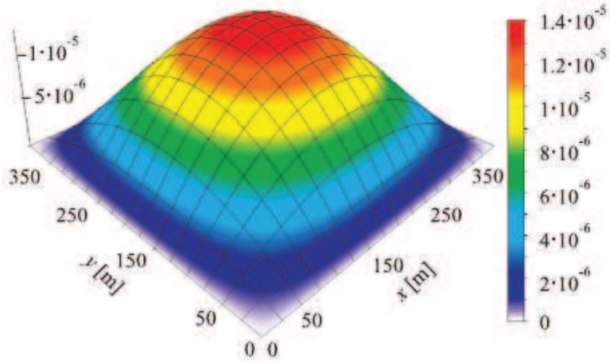


Figure 2.8: Density for the random-waypoint model (taken from [95])



$$\varphi_i = \frac{(k_i+1)^\alpha}{\sum_j (k_j+1)^\alpha} \tag{2.1}$$

$$where:$$

$$\varphi_i\, is\, the\, probability\, that\, a\, node\, selects\, subarea\, s_i\, as\, destination$$

$$\alpha\, is\, the\, clustering\, exponent$$

$$k_i\, is\, then\, umber\, of\, nodes\, in\, subarea\, s_i$$

$$s_i\, is\, a\, subarea\, with\, 0 < i \leq s_t\; (s_t\, is\, the\, total\, number\, of\, nodes)$$

When the growth phase is finished, φ_i is calculated and it will not be changed anymore in the *rewiring* phase up to the simulation has finished. Here is time to give mobility to nodes, selecting a sub area based on φ_i and a position inside as its next destination. Node selects a speed uniformly distributed in $[v_{min}, v_{max}]$ (with $v_{min} > 0$ in order to mitigate *speed decay* [97]) and starts to move

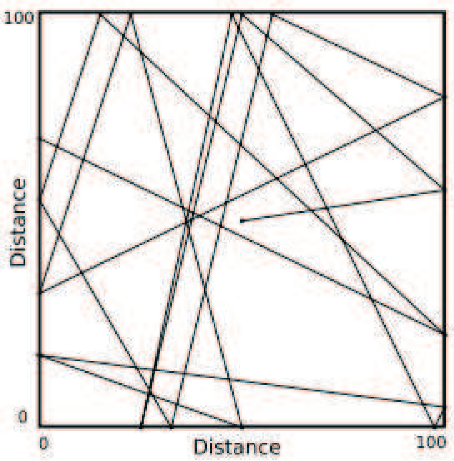


Figure 2.9: RDM movement pattern for one node

towards it.

This model allows to control the non-uniform phenomenon (not like in *RWP*), by the use of cluster with probability function φ_i and to tune it by the clustering exponent (α). Lim et al also conclude that it reaches the steady-state in less time than other random models.

2.4.2 Temporal dependencies

Now is time to describe the models where a node movement depends on the past movements of the node.

2.4.2.1 A probabilistic version of Random Walk

Chiang proposed a modified version of *RWM* [31] to provide a more stable (and smooth) node movement (Figure 2.13 on page 18), controlling movements by a three-state Markov chain as shown in Figure 2.12 on page 17, in which a state-transition means changes in X-direction and Y-direction (both in state (0) initially). This model is implemented as a probability matrix:

$$P = \begin{bmatrix} P(0,0) & P(0,1) & P(0,2) \\ P(1,0) & P(1,1) & P(1,2) \\ P(2,0) & P(2,1) & P(2,2) \end{bmatrix}$$

where $P(i,j)$ denotes the probability that a node will go from state i to state j. By tuning the probability values on this Markov chain, different patterns may be modeled, and temporal dependency could be handled, giving more or less probability to transitions.

2.4.2.2 Boundless Simulation Area mobility model

This model introduces a novel approach about the simulation area, in which once a node reaches a boundary, it wraps around to the other side of the area [26, 50], mapping a rectangular simulation area in a torus (Figure 2.14 on page 18).

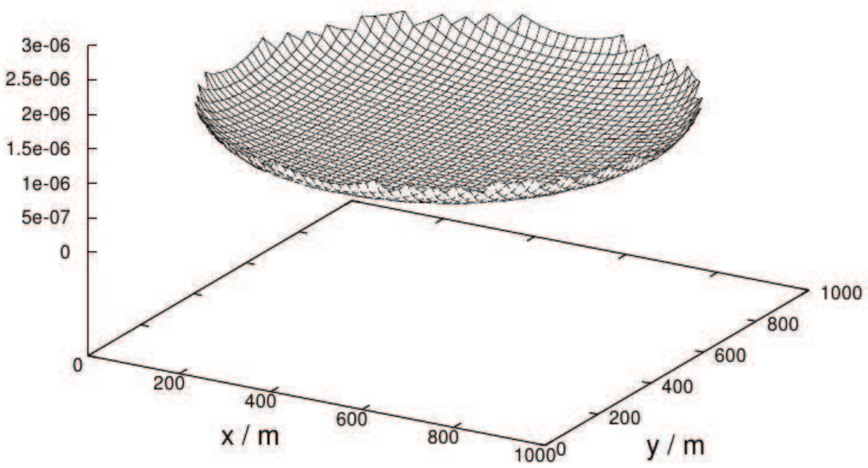


Figure 2.10: Node distribution in circular area (taken from [14]).

This modification tries to remove the *non-uniform node distribution* phenomenon, present in the random models (see Figure 2.15 on page 19). With this model, movements of nodes have temporal dependencies, making present *direction* of travel and *speed* dependent of the previous ones [96].

Movements of a node are guided by a velocity vector $\vec{v} = (v, \theta)$ which describes node velocity v and its direction θ, and the node position (x, y) is updated every Δt time steps, as follow:

$$v(t + \Delta t) = min[max(v(t) + \Delta v, 0), v_{max}];$$

$$\theta(t + \Delta t) = \theta(t) + \Delta\theta;$$

$$x(t + \Delta t) = x(t) + v(t) * cos\theta(t);$$

$$y(t + \Delta t) = y(t) + v(t) * sin\theta(t);$$

v_{max} is the maximum velocity defined, and v is the change in velocity uniformly distributed between $[-A_{max} * \Delta t, A_{max} * \Delta t]$. A_{max} is the maximum acceleration/deceleration for a node, $\Delta\theta$ is the change in direction uniformly distributed between $[-\alpha * \Delta t, \alpha * \Delta t]$, and α is the maximum angular change in the direction for a node.

Even when this type of models seems to fit well for certain scenarios (e.g. railway problem where trains run on rails without collisions, correlation of movement to technical aspects [10]), and it removes restrictions about simulation borders and their effects, it has its own disadvantages. Authors in [26] note that *"undesired side effects that would occur from allowing the MNs to move around a torus. For example, one static MN and one MN that continues to move in the same direction become neighbors again and again. In addition, a simulation area without edges would force modification of the radio propagation model to wrap transmissions from one edge of the area to the other"*.

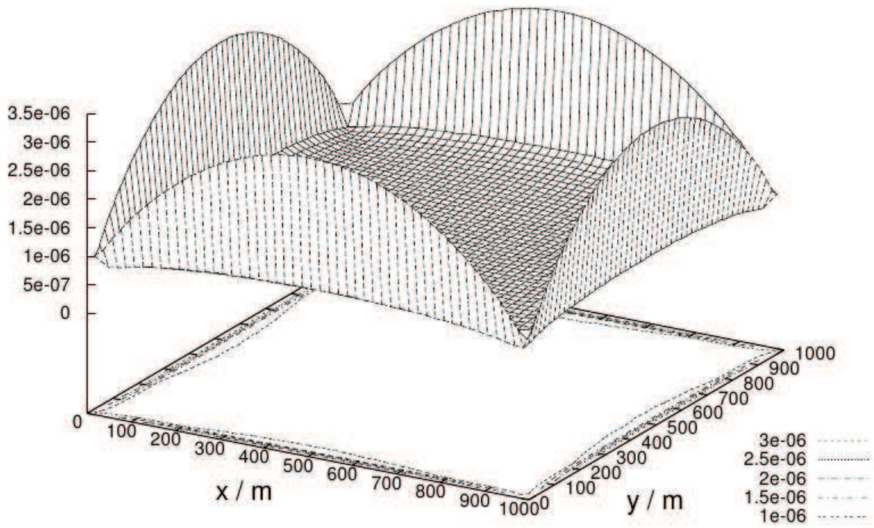


Figure 2.11: Node distribution in a square area (taken from [14]).

2.4.2.3 Gauss-Markov mobility model

Close to the *Random Waypoint* model is the *Gauss-Markov model* [77], which includes some enhancements to the resulting path to adapt different levels of randomness via one tuning parameter. This model adds a kind of memory to each mobile node, so decisions at the waypoint are made taking into consideration past velocities and direction values (in fixed intervals of time[1]) in the following way:

$$s_n = \alpha s_{n-1} + (1-\alpha)\vec{s} + \sqrt{(1-\alpha^2)}s_{x_{n-1}} \tag{2.2}$$

$$d_n = \alpha d_{n-1} + (1-\alpha)\vec{d} + \sqrt{(1-\alpha^2)}d_{x_{n-1}} \tag{2.3}$$

where s_n and d_n are the new speed and direction of node at time interval n, $\alpha \in [0,1]$ is the randomness parameter, $\vec{s}$ and $\vec{d}$ are the speed and direction mean values as $n \to \infty$ and $s_{x_{n-1}}$ and $d_{x_{n-1}}$ are random variables from a Gaussian distribution.

Derived from previous equations, the calculation of next node position is as follow:

$$x_n = x_{n-1} + s_{n-1}cosd_{n-1} \tag{2.4}$$

$$y_n = y_{n-1} + s_{n-1}sind_{n-1} \tag{2.5}$$

where (x_n, y_n) and (x_{n-1}, y_{n-1}) are the node position at time interval n and $n-1$ respectively,

[1] Original study made derivation for continuous-time, but given some practical limitations, cars use discrete time intervals.

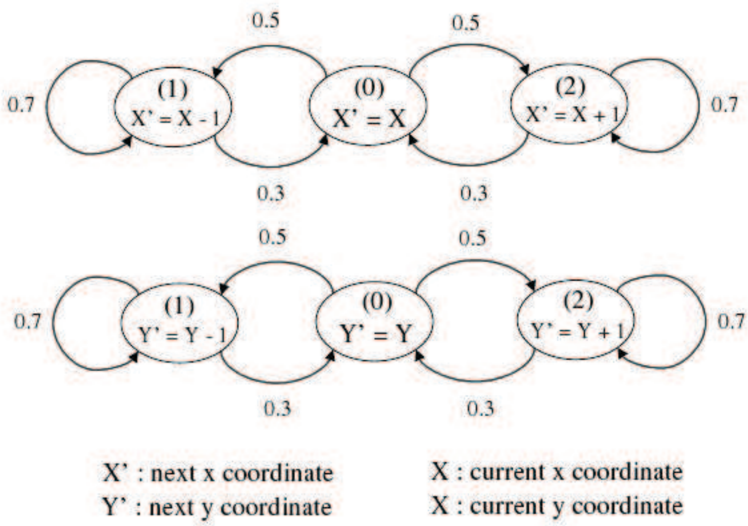


Figure 2.12: Markov chain for location transitions (taken from [31]).

and speed s_{n-1} and direction $d_{n-1}$are the correspondent speed and direction at time interval $n-1$.

One modification introduced in [128] for the *Gauss-Markov* model define a mechanism to avoid that a node remains for long time near the edge, that consists in changing mean direction $\vec{d}$. An example of this mechanism is depicted in [26], that is only applied when a node is within some distance of the edge as its shown in Figure 2.16 on page 20.

Correlations between past and present speed and direction eliminate the sudden stop and sharp turns encountered in random mobility models (see Figure 2.17 on page 21).

As consequence of the randomness tuning parameter α, these models behave in different ways, depending on the value of it according with (2.3) and (2.2):

- If $\alpha = 0$, the *Gauss-Markov* model is a memoryless model similar to the *Random Walk* model, and equations remains as $s_n = \vec{s} + s_{x_{n-1}}$ and $d_n = \vec{d} + d_{x_{n-1}}$
- If $\alpha = 1$, *Gauss-Markov* model is totally dependent on last past values of speed and direction (are the same), and equations remains as $s_n = s_{n-1}$ and $d_n = d_{n-1}$. This model is called as fluid flow model, in the nomenclature of vehicular traffic theory.
- Varying α between $(0, 1)$ it is possible to get different levels of randomness, making that next positions more dependent on the lasts speed and direction, or more randomly chosen from $s_{x_{n-1}}$ and $d_{x_{n-1}}$.

2.4.2.4 Smooth Random mobility model

This mobility model defined in [11] consider a temporal dependency of velocity over various time slots, and is based on (or is an enhancement of) *Random Direction model. Random Waypoint model* presents sudden stops, acceleration/deceleration, and sharp turns, characteristics that *Smooth*

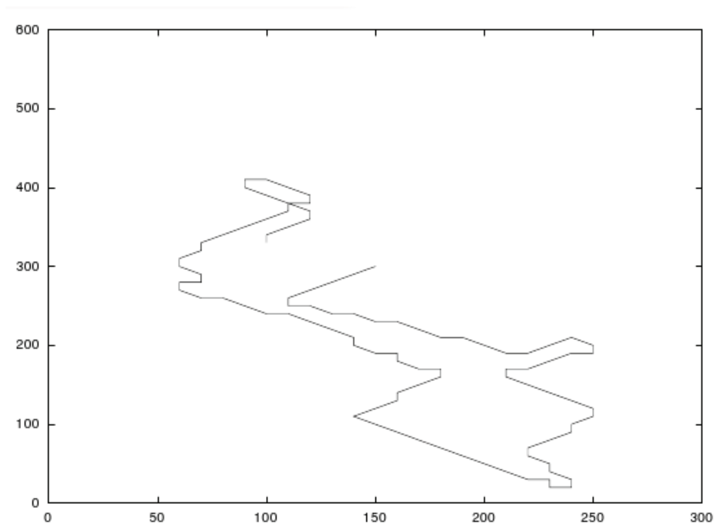


Figure 2.13: Movement pattern for probabilistic RWM (taken from [26]).

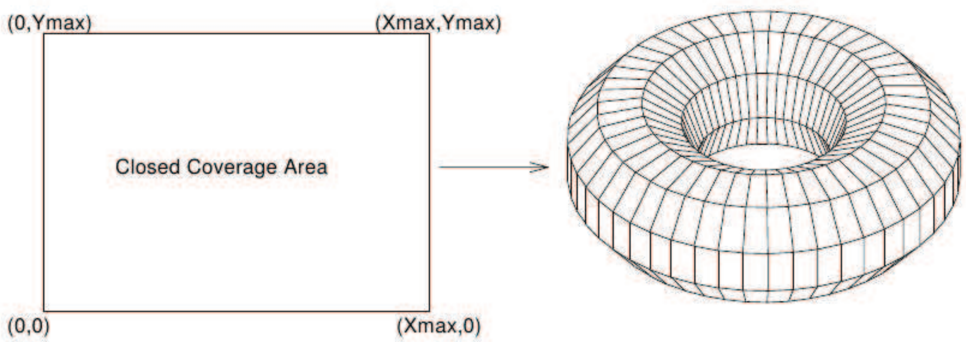


Figure 2.14: Mapping of simulation area (taken from [26]).

Random model tries to convert in a more realistic manner, i.e. making changes in speed and direction incrementally and smoothly (see an example in Figure 2.18 on page 22).

This model uses two stochastic processes: one to determine when a node changes its speed, and the second at what time the direction will be changed. These changes are autocorrelated, speed changes incrementally by the node acceleration, and direction change is smooth, because once a node begins to turn, direction is changed in several time slots until target direction is achieved.

Based on the observation that speed is not uniformly distributed, a set of preferred speeds $\{V^1_{pref}, V^2_{pref}, \ldots, V^n_{pref}\}$ with major probability are used (and uniformly distributed for the rest), instead of a speed uniformly distributed for all intervals $[0, V_{max}]$. For example, a probability distribution of node velocity with a set of speed $\left\{0, \frac{3V_{max}}{5}, V_{max}\right\}$ is as follow:

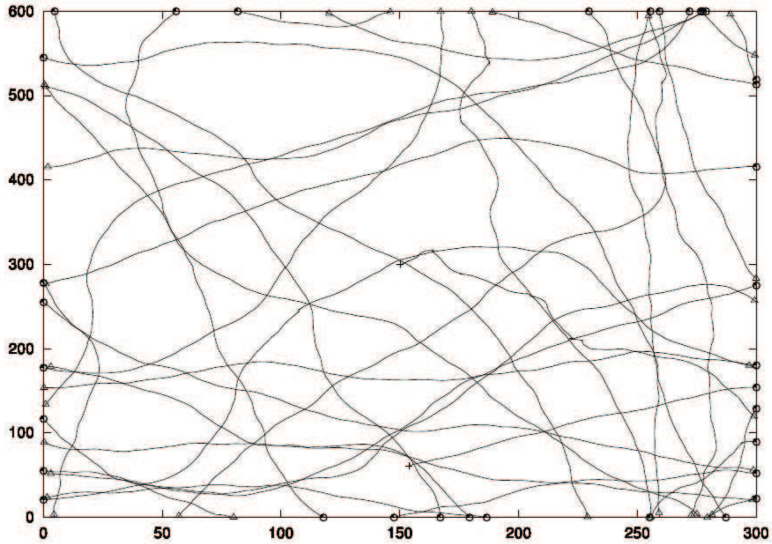


Figure 2.15: Node movement in Boundless simulation area(taken from [26]).

$$p(v) = \begin{cases} p(v=0)\delta(v) & v=0 \\ p(v=\frac{3V_{max}}{5})\delta(v-\frac{3V_{max}}{5}) & v=\frac{3V_{max}}{5} \\ p(v=V_{max})\delta(v-V_{max}) & v=V_{max} \\ \frac{1-p(v_{pref})}{V_{max}} & 0<v<V_{max} \\ 0 & else \end{cases} \tag{2.6}$$

with $p(v_{pref}) = p(v=0) + p(v=\frac{3V_{max}}{5}) + p(v=V_{max}) < 1$.

The frequency of speed change is a Poisson process, and when a speed change event is raised, a new speed is chosen using Equation 2.6. If $v(t)$ is the current speed at time t and $v(t')$ the targeted new speed, mobile node will change its velocity ruled by an acceleration/deceleration $a(t)$ taken with uniform distribution from $[0, a_{max}]$ and $[a_{min}, 0]$ if $v(t')$, is major or minor respectively than $v(t)$. In consequence, the new speed for each time slot is calculated as $v(t) = v(t-\Delta t)+a(t)\Delta t$.

Directions, in contrast with speed, are selected uniformly distributed between $[0, 2\pi]$, and the frequency of direction changes are modeled as an exponential distribution. When a change of direction event is raised, a new direction $\phi(t')$ is chosen, and the node will achieve that direction by issuing successive and incremental changes every Δt time steps, from its actual direction $\phi(t)$ and until $\Delta\phi(t)$ (the difference of actual and new direction) angle is complete. The value $\Delta\varphi(t)$ should be small, and it represents the maximum direction change in a time slot, so targeted direction will be achieved in $\frac{\Delta\phi(t)}{\Delta\varphi(t)}$ time slots. After that, the node continues to move in the targeted direction.

As a matter of summary, this model tries to enhance Random Direction model, generating smooth movement patterns, and giving a way to tune it varying the two parameters $\Delta\varphi(t)$ and acceleration, in order to adjust the degree of temporal dependency and its impact.

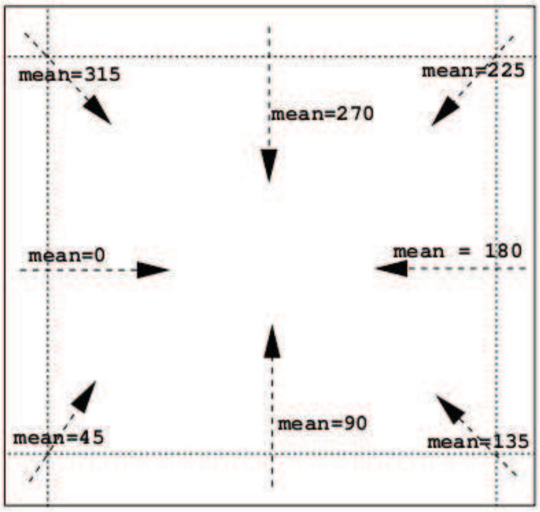


Figure 2.16: Change of mean angle near edges, in degrees (taken from [26]).

2.4.2.5 Random Trip mobility model

Random trip model [73, 74] provides a framework to analyze and simulate stable mobility models that are guaranteed to have a unique time-stationary distribution. Moreover, conditions are provided that guarantee convergence in distribution to a time-stationary distribution, from origin of an arbitrary trip. This is a generic mobility model for random and independent node motion, and its goals are (i) to provide a class of "stable" mobility models that is rich enough to accommodate a large variety of examples and (ii) to provide an algorithm to run "perfect simulation" of these models.

The model is defined by a set of "stability" conditions for a node movement that guarantee existence of a time-stationary regime of node mobility state or its non existence. They also guarantee convergence of node mobility state to a time-stationary regime, whenever one exists, starting a node movement from origin of a trip. In summary, this model tries to generate simulations for different scenarios with the idea of reaching a steady state simulation, avoiding problems like speed decay, speed decaying to 0, etc, or having conditions to know if a stable state will be reached in the future.

Even when stability conditions are met, simulations necessarily stay in a transient state for certain time until it converges to a stationary regime. In a transient period of the simulation, measurements could be unusable, and in consequence they should not be taken in consideration, so its necessary to wait until steady-state is reached (if simulation run is long enough).

For stable random trip models, if the initial node mobility state is not sampled from the time-stationary distribution, the node mobility state distribution converges to the time-stationary distribution. The rate of this convergence depends on the geometry of the mobility domain and specific of the trip selection. In order to alleviate this initial transience altogether, Le Boudec et al provide a perfect sampling algorithm to initialize node mobility state to a sample from the time-stationary distribution [89].

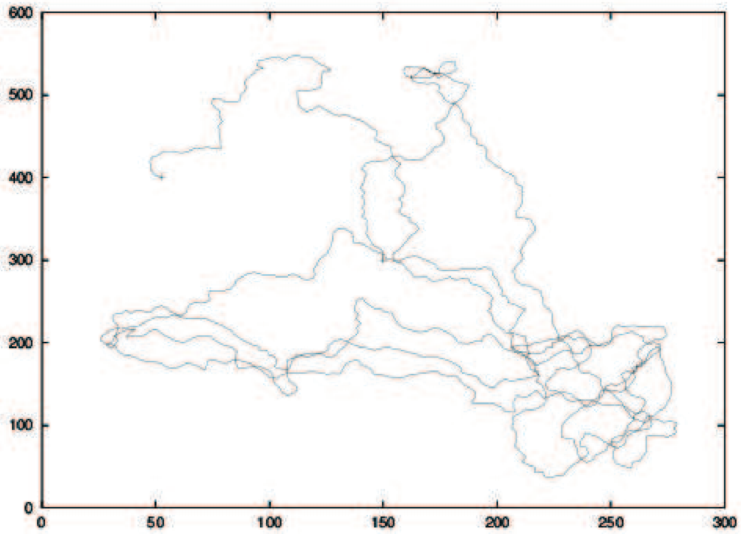


Figure 2.17: Node's moving patter in Gauss-Markov model (taken from [26]).

2.4.3 Spatial dependencies

This section describes the models that movements of a node depends on the nodes around.

2.4.3.1 Exponential Correlated Random mobility model

This is one of the first group mobility models, defined in [82], which has a motion function used to generate movement patterns. As described in [26], for a node in a given position $\vec{b}(t)$ at time t, next position at time $t+1$, $\vec{b}(t+1)$ is:

$$b(t+1) = b(t)e^{\frac{-1}{\tau}} + (\sigma\sqrt{1-(e^{\frac{-1}{\tau}})^2})r$$

where τ adjusts the rate of change from old to new (τ small causes large change); r is a random Gaussian variable with a variance σ. The parameters τ and σ vary from group to group, and they derive different moving patterns for each group or node, and for this reason this model requires a complete set of (τ, σ) ; (one per group) to define the motion of the entire network. For Bergamo et al the drawback is that it is not easy to force a given motion pattern by selecting the parameters.

2.4.3.2 Reference Point Group mobility model

Reference Point Group Mobility model (RPGM) was defined in [53] as an effort to mitigate problems with previous work in this kind of models and to better reflect interactions between nodes by the proper choice of a set of parameters. This model is the most general group mobility model, and it has been used as a framework for the implementation of several others as a special case of it.

Mobile nodes are organized in groups, and each group has a logical entity called center (or so-called *group-leader*), which directs the behavior of the group, compound by nodes that are called

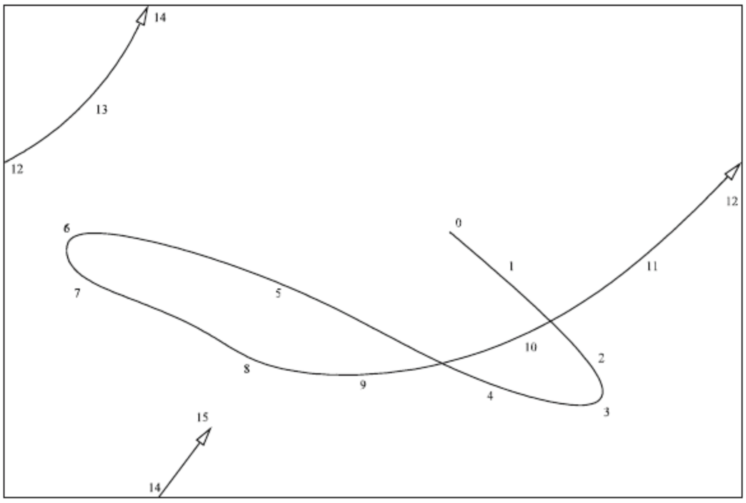


Figure 2.18: Movement pattern for a node (taken from [121]).

group-members, so mobile nodes will be seen as a group of nodes moving together. Movement of group-members are not only guided by the group-leader, but they include an independent movement, in addition to the group motion.

Movement of nodes inside a group could be described as follows (see Figure 2.19 on page 23):

- First, a reference point is moved according to a $\vec{GM}$ vector (each group has its own GM vector), so node position is updated from $\vec{RP}(t)$ to $\vec{RP}(t+1)$.
- After that, a new node position is calculated adding a $\vec{RM}$ vector (each node has its own RM vector) to the new reference point $\vec{RP}(t+1)$. The length of $\vec{RM}$ is uniformly distributed within a certain radius centered at the reference point and its direction uniformly distributed between $[0, 360]$ degrees.

RPGM explicitly defines the motion path for the reference point and for each group, by giving a sequence of check points and the time in which the reference point will arrive to them. Each time a reference point reaches the next check point, the calculations of new node positions are made. Due to the fact that motion paths are predefined, Hong et al highlight that *"the model has the advantages of providing a general and flexible framework for describing mobility patterns, which are task oriented and time restricted as well as easy to implement and verify "*.

As example of the flexibility of this model, based on proper choice of motion paths and radio for reference point, several mobility models are derived from it (and they were defined in [53]):

- *In-place Mobility Model*: Area is divided in adjacent regions, and a group is placed inside of each one. This can be used to model situations in which groups are doing same operations at the same time in different subareas.

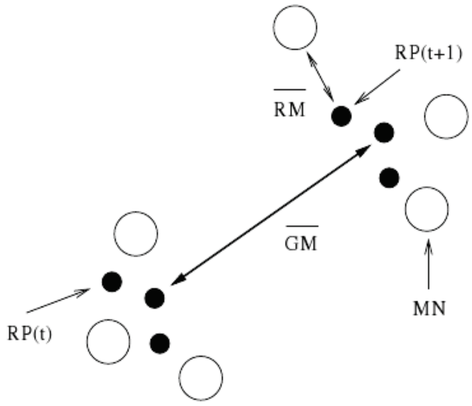


Figure 2.19: RPGM movement pattern for 3 nodes (taken from [26])

- *Overlap Mobility Model*: *"The second model describes an overlapped operation. Different groups carry out different tasks over the same area. However, the distinct requirements of each task make their mobility pattern quite different"* [53].
- *Convention Mobility Model*: It models interactions in a convention, reflecting the fact that some groups move from one room to another, or stay in one for certain time and then continue.

A related work is the *Mobility Vector framework* introduced in [6], based on Mobility Vector Model [69]. In this framework it is possible to model various scenarios too (like *Gravity Model*, *Targeting Model*, *Location Dependent Model* and *Group Motion Model*), choosing the proper parameters. In this model, each node has its own *Mobility vector* $\vec{M} = (x_v, y_v)$, and it represents the positions for each time unit, so velocity could be derived by a simple calculation using two consequent positions between current time and the current time plus a time unit. *Mobility vector* is composed by two sub vectors:

- *Base Vector* $\vec{B} = (bx_v, by_v)$ which defines major direction and speed of a node
- *Variable Vector* $\vec{V} = (vx_v, vy_v)$ that defines derivation from $\vec{B}$

resulting in $\vec{M} = (x_v, y_v) = \vec{B} + \vec{V} = (bx_v + vx_v, by_v + vy_v)$

Additionally, Bai et al define some properties: *Minimum/Maximum Speed*, *acceleration/deceleration factor* and mention the possibility of defining a *Min/Max angle* and the *steering factor* to get direction changes more naturally (in a similar way than *RDM* or *Smooth Mobility Model*).This extension was named *Natural Random Mobility.*

Another modification of *RPGM* is used in [6] with the objective to give a stronger spatial dependency to group-members. For this reason, speed and direction vectors are defined using not only a random motion vector, but using new parameters instead (2.7):

$$\begin{cases} |V_{member}(t)| = |V_{leader}(t)| + random() * SDR * maxspeed \\ |\theta_{member}(t)| = |\theta_{leader}(t)| + random() * ADR * maxangle \end{cases} \tag{2.7}$$

where *SDR (Speed Deviation Ratio)* and *ADR (Angle Deviation Ratio)* are used to control *spatial dependency* of nodes from the group-leader, by adjusting this two parameters, choosing them in $0 \leq SDR, ADR \leq 1$.

2.4.3.3 Column mobility model

This model is a special case of *RPGM*, applicable to scenarios like searching and scanning [126], in which a group of MNs move on a line and are uniformly moving forward in a particular direction. At the beginning of the simulation a reference grid is defined, forming a column of reference points. After that, MNs are placed in relation of their reference points: then MNs are allowed to move randomly (for example using *RWM*) around their reference point.

Movements are defined as follow: *new_ref_point* is the next reference point for a MN, *old_ref_point* is the previous reference point, and *advance_vector* is a predefined offset that moves the reference grid.

$$new\_ref\_point = old\_ref\_point + advance\_vector$$

The same predefined offset is used by all MNs, calculated via random distance and random angle between $[0, \pi]$ (because they are only allowed to move forward), in consequence the reference grid is a 1-D line (see examples in Figure 2.20 on page 24).

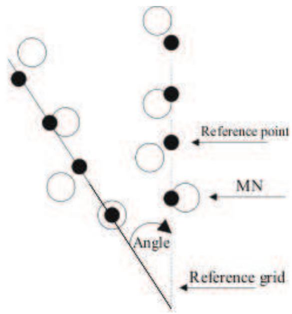


Figure 2.20: Movement example for Column model (taken from [26]).

2.4.3.4 Nomadic Community mobility model

This model is a special case of *RPGM*, and represent groups of MNs that *collectively* move from one point to another [26, 90, 126]. Within each community or group of *MNs*, individuals maintain their own personal *"spaces"* where they move in random ways. Each *MN* uses an entity mobility model (e.g., the *Random Walk Mobility Model*) to roam around a given *reference point.* When the reference point changes, all MNs in the group travel to the new area defined by the *reference point* and then begin roaming around the new reference point. Figure 2.21 show a movement pattern, in which black dot is the reference point, and *MNs* are represented as white balls.

In this model, parameters could be tuned for the entity mobility model, in order to define how far a MN may roam from the reference point. In comparison with *Column Mobility Model*, the MNs in the Nomadic Community Mobility Model share a common reference point versus an individual reference point in a column, thus, it is expected that MNs will be less constrained in their movement around the defined reference point.

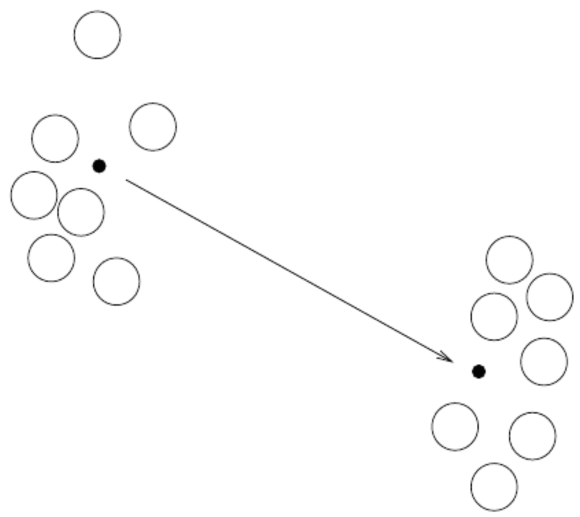


Figure 2.21: Movements of nodes in Nomadic Community Model (taken from [26]).

2.4.3.5 Pursue mobility model

As *Column* and *Nomadic Community* models, this model is defined in [26, 90, 126], and attempts to represent MNs tracking a particular target. The *Pursue Mobility Model* consists on a single update equation for the new position of each MN:

$$newposition = oldposition + acceleration(targetoldposition) + randomvector$$

where $acceleration(targetoldposition)$ is information on the movement of the MN being pursued and $randomvector$ is a random offset for each MN. The random vector value is obtained via an entity mobility model (e.g., the *Random Walk Mobility Model*); the amount of randomness for each MN is limited in order to maintain effective tracking of the MN being pursued. The current position of an MN, a random vector, and an acceleration function are combined to calculate the next position of the MN.

Figure 2.22 shows six MNs moving with the Pursue Mobility Model. The white node represents the node being pursued and the solid black nodes represent the pursuing nodes.

2.4.3.6 Reference Velocity Group mobility model

Reference Velocity Group Model (RVGM) [117, 118] is included in a study of characterization of group mobility based on existing group mobility models, which provides parameters that are enough for network partition prediction. Radhika Ranjan demonstrate how partition prediction can be made using the mobility model parameters, and he illustrates the applicability of the prediction information. Using a simple but effective data clustering algorithm, given the velocities of the mobile nodes in an ad-hoc network, the model could accurately determine the mobility groups and estimate the characteristic parameters of each group.

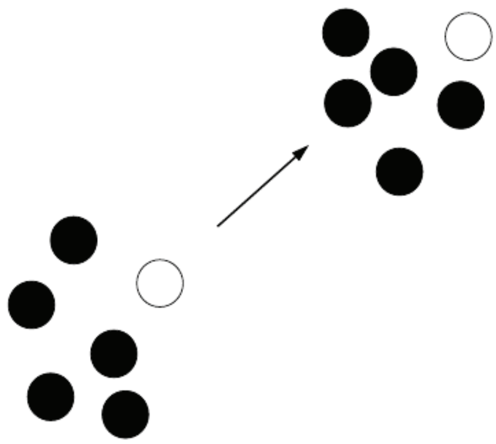


Figure 2.22: MNs moving with Pursue Mobility Model (taken from [26]).

RVGM is an extension of RPGM, that tries to create a more accurate group movement based on the observation that instead of proximity in physical displacements, a more fundamental characteristic of a mobility group is the similarity of the member nodes' movements. Therefore, authors of *RVGM* proposed a velocity representation of the mobility groups and the mobile nodes: Each mobility group has a characteristic *group velocity*. The member nodes in the group have velocities close to the characteristic *group velocity* but deviate slightly from it. Hence, the characteristic *group velocity* is also the *mean group velocity*.

As another extension, the Reference Velocity and Acceleration Group Mobility model is proposed in [30]. In this model not only a velocity group is taken into consideration for node movement, but also the acceleration.

2.4.3.7 Structured Group mobility model

Structured Group Mobility Model (SGMM) [17] is an extension of *RPGM* aimed to model group mobility with major accuracy. Existing models show movement of individual nodes within groups with random movements. *SGMM* extends group mobility models by incorporating a-priori knowledge of structure into the movement of groups of nodes. In *SGMM*, individual nodes are assigned to groups according to a known organizational structure, and these groups move in concert with other groups in a larger operation. Individual nodes maintain stable relationships within their group, thus preserving overall group structure.

Blakely and Lowekamp suggest that mobility task orientation exists within a group (because of a common goal) and that this orientation will be known to the individual creating the simulation beforehand. The *SGMM* is shown in Figure 2.23 and it is defined as follows:

- each group j has a reference point c_j (geographical center of the group, the location of the leader, or the group's center of mass)

- c_j has a directional orientation of angle T from 0 degrees on a global coordinate system. Thus c_j is able to maintain an orientation independent of movement, and subordinate groups and nodes are positioned relative to c_j
- Each subordinate groups or node i occupies a location relative to c_j
- The model derives this position by selecting a distance d_i from c_j from a given distribution D and an angle a_i away from T from a given distribution A
- The relationship between each node and the reference point is maintained by specifying the distributions D and A from which d_i and a_i are selected

Since all subordinate nodes are dependent on c_j for their position, defining the movement of c_j is enough to define the movement of the entire group. Thus, in group j, the location of node i is updated over time based on four parameters, and the location at time t is calculated as:
$i(t) = F(c_j, T, D_i, A_i)$

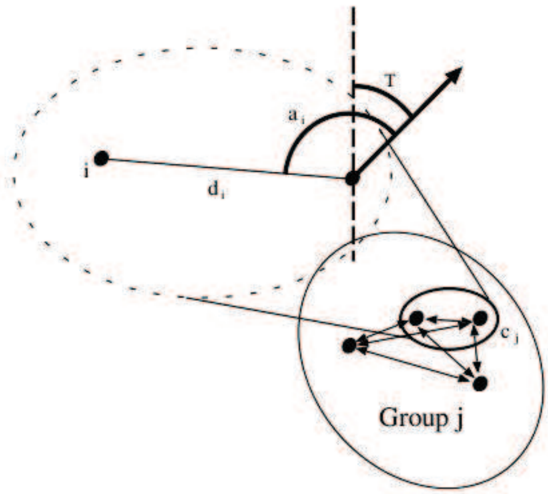


Figure 2.23: SGMM: Placement of node i (taken from [17]).

2.4.3.8 Others group models

A combination of group and entity models was proposed in [138] and was called *Two-tier mobility model*. This model is based on correlation of mobility states and is more general than other group mobility models. The first tier of the model represents the mobility state evolution of an individual node while the second tier represents the interactions between the mobility states of nodes belonging to a group.

The second tier of the model is invoked only when there is enough evidence of group behavior. Authors propose the use of a *correlation index test* to determine the presence of correlation between the mobility states of nodes. The correlation index ρ is a normalized parameter which has the value 0 for uncorrelated random variables. A value of ρ close to 1 or -1 indicates strong correlation between the variables.

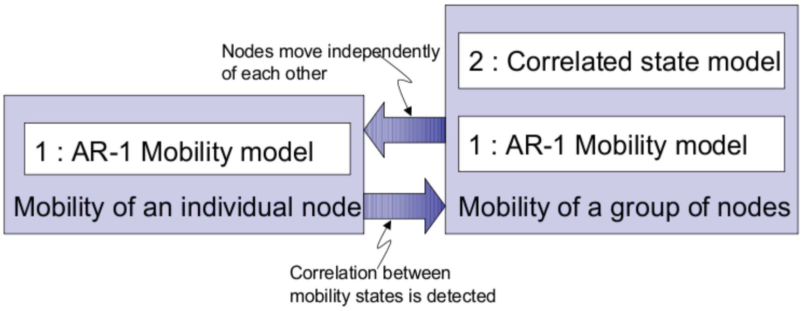


Figure 2.24: Two-tier mobility model overview (taken from [138]).

An example of mobility for testing application is [120] in which worst case scenarios studies were made. In this model, the pedestrians mobility assumes a maximum speed v_{max} (*velocity bounded model*) and the vehicular mobility assumes a maximum acceleration a_{max}(*acceleration bounded model*).

2.4.4 Geographic restrictions

The models described in this section are those models on which not all places in the area are allowed to be transited.

2.4.4.1 Freeway mobility model

Freeway Mobility Model (FWMM) [6] emulate the motion behavior of mobile nodes on a freeway (see Figure 2.25), and it can be used in exchanging traffic status or tracking a vehicle on a freeway. In this model maps are used, and there are several freeways on the map and each freeway has lanes in both directions. The velocity of each mobile nodes is temporally dependent on its previous velocity, and even more, if two mobile nodes on the same freeway lane are within the Safety Distance (SD), the velocity of the following node cannot exceed the velocity of preceding node. *Inter-node* and *intra-node* relationships are:

$a)\,|\vec{V_i}(t+1)| = |\vec{V_i}(t)| + random() * |\vec{a_i(t)}|$
$b)\,\forall i, \forall j, \forall t\ D_{i,j}(t) \leq SD \Rightarrow |\vec{V_i}(t)| \leq |\vec{V_j}(t)|$ if j is ahead of i in its lane

2.4.4.2 Manhattan mobility model

Manhattan Mobility Model [6] is commonly used by simulations of urban areas, where mobile nodes moves in a grid in for directions (up, down, left and right) as in Figure 2.26. When a mobile node gets a corner of the grid, it takes a decision with some probability upon which direction it will take at next. The velocity of a mobile node at a time slot is dependent on its velocity at the previous time slot. Also, a node's velocity is restricted by the velocity of the node preceding it on the same lane of the street. The *inter-node* and *intra-node* relationships involved are the same as in the *Freeway model*. Thus, the *Manhattan mobility model* is also expected to have high spatial dependence and high temporal dependence. It too imposes geographic restrictions on node mobility. However, it differs from the *Freeway model* in giving a node some freedom to change its direction.

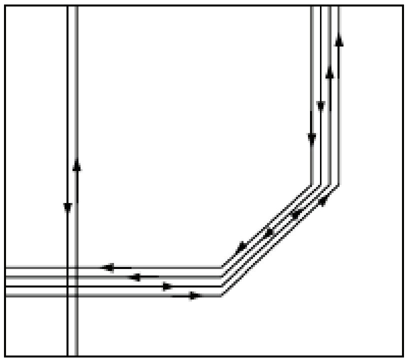


Figure 2.25: Freeway scenario example (taken from [6]).

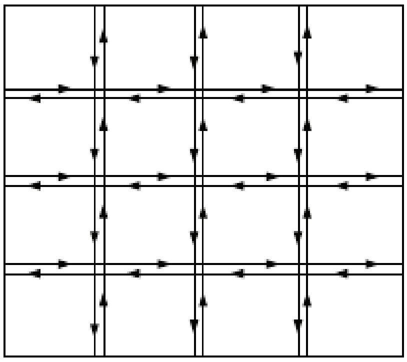


Figure 2.26: Manhattan grid map example (taken from [6]).

2.4.4.3 City Area, Area Zone, and Street Unit models

In [85], the authors take an in-depth look at desirable characteristics of mobility models including required inputs/outputs and issues that should be considered when designing a specific mobility model. They proposed three basic types of mobility models, which are appropriate for the analysis of the full range of mobile communications design issues. Each model provides different levels of detail regarding the user mobility behavior. In particular: (a) the *City Area Model* traces user motion at an area zone level, (b) the *Area Zone Model* considers users moving on a street network and (c) the *Street Unit Model* tracks user motion with an modeling location management in personal communications services accuracy of a few meters.

The general idea of each of these models is to have a set of input parameters (population P, which represent specific groups of MNs, a geographical area G organized into regions, and a time period T), a set of output parameters (a collection of functions that determine the location of an MN p over the set G at time t), and the use of transportation theory in order to get output parameters from input ones. Transportation theory aims to resolve the following: "Given a transportation system serving a certain geographical area, determine the load this system should carry" and its main describing items are:

- *Trips*: A trip of a node.

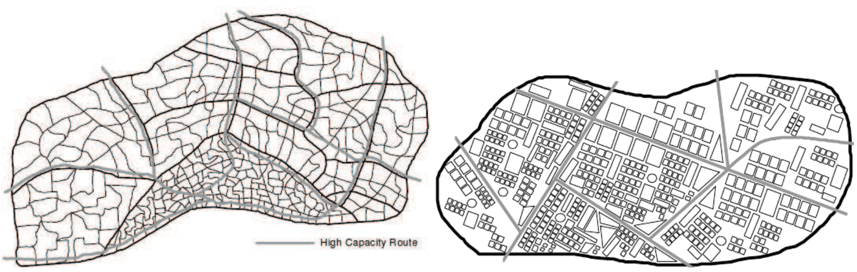


Figure 2.27: (left) City Area and (right) Area Zone (taken from [85]).

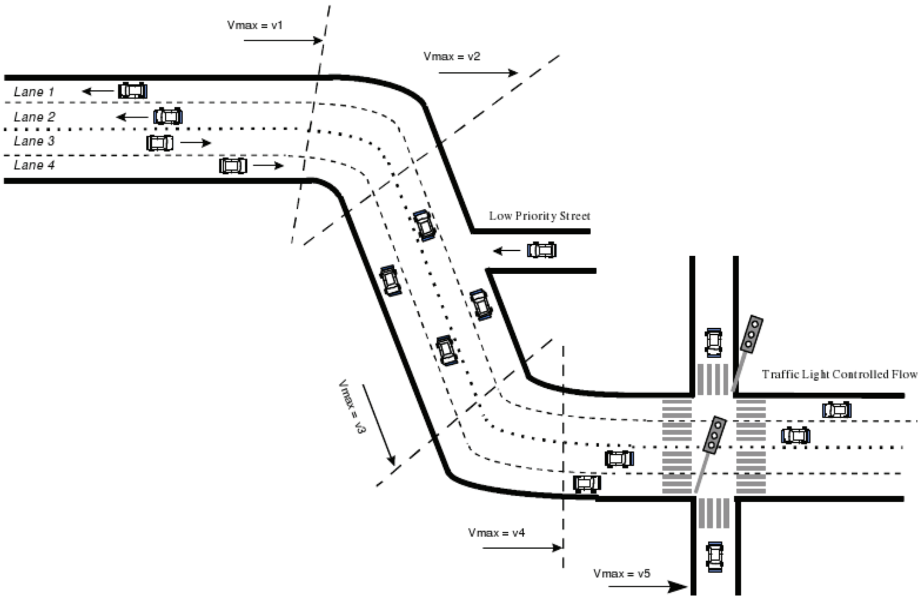


Figure 2.28: Street Unit Model (taken from [85]).

- *Area Zones*: The geographical area under study is divided into area zones (based on (a) the population density and (b) the natural limits).
- *Population Groups*: The population is divided according to their mobility characteristics (working people, residential users, students, etc).
- *Movement Attraction Points (MAP)*: Represent locations that attract the population movements and at which people spend considerable time periods.
- *Time Zones*: During a day time, it can be observed that there are time periods during which certain types of movements take place and time periods where certain population groups reside to certain MAPs. Transportation theory concentrates on the so-called ‘rush hours’, where the peak load occurs on the transportation system under study.
- *Transportation Systems Characteristics*: A transportation system (e.g., a street network, the urban buses network, the subway, etc.) is characterized by: (a) its capacity, (b) the trips it may support and (c) the usage cost.

Markoulidakis et al presents a novel tool, called *Integrated Mobility Modeling Tool (IMMT)*, in which these sub-models are components of a major model and interact between them. An overview of structure and interactions are depicted in Figure 2.29.

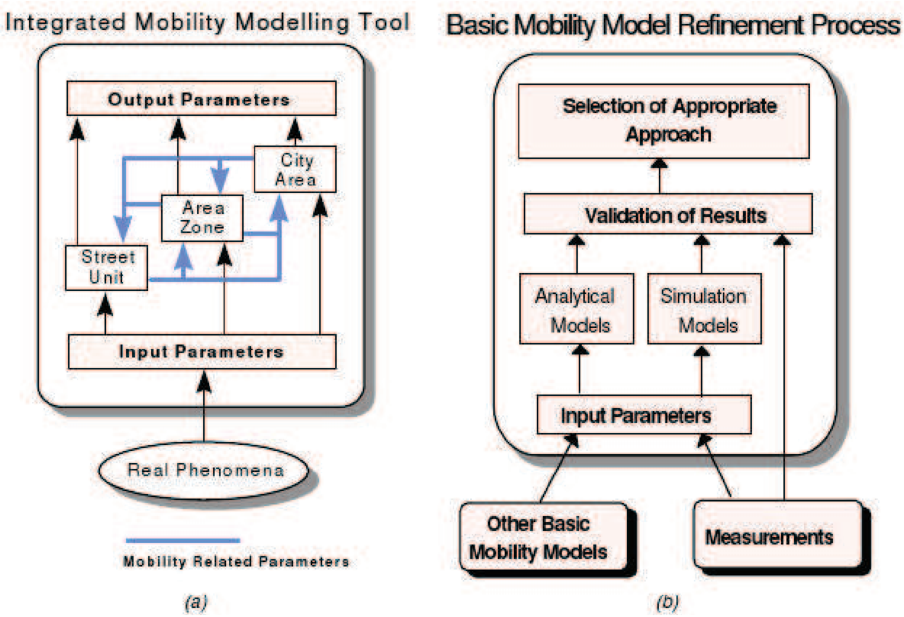


Figure 2.29: (a) IMMT logical view and (b) Refinement process (taken from [85]).

Related to this work, the hierarchy of mobility models could be extended in scale, like in [70], where Lam et al had proposed models and these models were simulated with Pleiades [71], which is a simulator developed by Lam et al:

- The *Metropolitan Mobility Model (METMOD)* describes the subscriber movements within a metropolitan area. It is a detailed model that includes the Markovian model as a special case.

- The *National Mobility Model (NATMOD)* characterizes movement behavior between metropolitan areas in the United States. Each site object now represents a metropolitan area.

- The *International Mobility Model (INTMOD)* characterizes movement behavior between the U.S. and ten other countries. Authors used a variant of the gravity model following the same methodology as in *NATMOD*.

2.4.4.4 City Section mobility model

In the *City Section Mobility Model* [34], the simulation area, represented by a grid, symbolizes horizontal and vertical streets within a city (see Figure 2.30 on page 32). Within the simulation environment, the centermost vertical and horizontal streets are designated as mid-speed roads, similar to main thoroughfares within a city. All other roads are considered to be slow residential roads. Each MN begins the simulation at a predefined intersection of two streets. A MN then randomly chooses a destination, also represented by the intersection of two streets. Moving to this destination involves (at most) one horizontal and one vertical movement, and locates a path corresponding to the shortest travel time. Upon reaching the destination, the MN pauses for a specified time, and randomly chooses another destination (i.e., an intersection of two streets) and repeats the process.

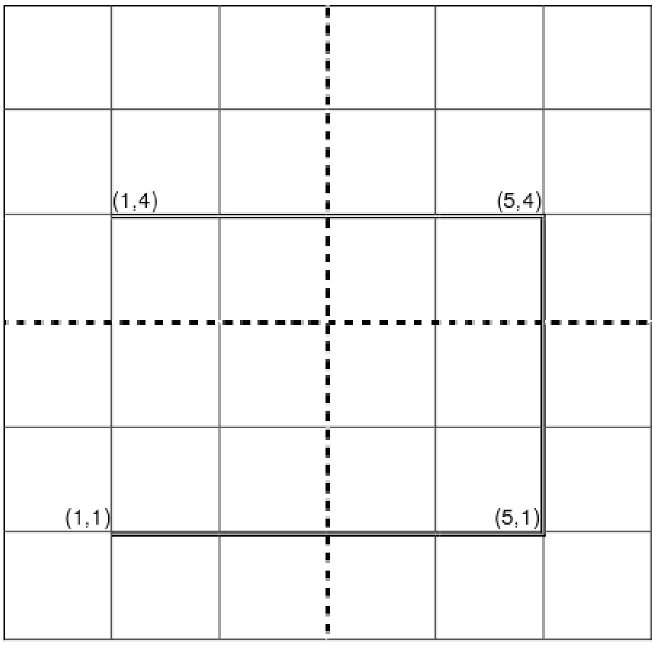


Figure 2.30: City Section movement pattern (taken from [34]).

As mentioned in [26] the *City Section Mobility Model* provides realistic movements for a section of a city since it severely restricts the traveling behavior of MNs. In other words, all MNs must follow predefined paths and behavior guidelines (e.g. traffic laws). At the same time, improvements to this model include pause times at certain intersections and destinations, incorporating acceleration and deceleration, and accounting for higher/lower concentrations of MNs depending on the time of day. In addition, the model should be expanded to include a larger simulation area, an increased number of streets, a high-speed road along the border of the simulation area, and other novel path-finding algorithms.

2.4.4.5 Weighted Waypoint mobility model

Weighted Waypoint (WWP) model [56] investigates the issue of non-uniform weights distribution (preferences) in choosing destinations and location dependent behavior of mobile nodes. Authors proposed to study the underlying mobility pattern directly, in complimentary to access point (AP) trace based study. They argue that the underlying mobility pattern will not change significantly for a given environment (e.g. a university campus), in contrast with the methodology used in [15].

The major differences of *WWP* model and the popular *RWP* model are: (a) *MNs no longer randomly chooses its destination*: this feature is modeled by defining popular locations in the simulation area and assigning different *"weights"* to them according to the probability of choosing destination from the area. In an instance of the *WWP* model, weights can be assigned by evaluating relative popularity of locations in the environment to be modeled; (b) *The "weights" of choosing next destination location depends on both current location and time*: a Markov model is used to capture this location-dependent weight assignment. If there are totally M locations in the simulation area, the weights can be represented using an $M \times M$ matrix. If time-dependency is also considered, then a time-variant matrix is obtained; and (c) *The pause time distribution at each location is different and is a property of that location.*

2.4.4.6 Restricted Random Waypoint mobility model

Restricted Random Waypoint (RRWP) is defined in [18] and its main characteristics are: (i) the area is divided in sub-areas (or towns), and (ii) the towns are connected by highways. Inside an area, nodes move with *RWP*, and after some number of movements in the same town, the node moves to another town. It reflects that in a large network, it is less probable that, for each movement, a node selects a random destination within a very large geographic area.

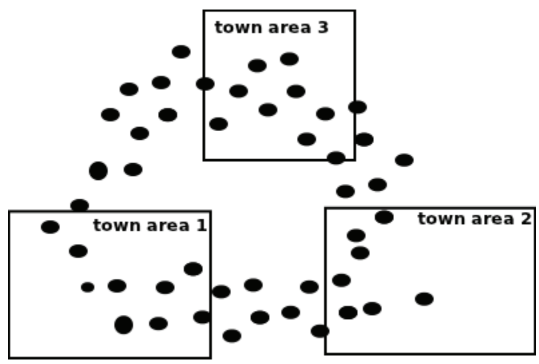


Figure 2.31: Topology example for RRWP

In the model, an *ordinary* node starts in a town, selects at random a destination within a new town, moves there with a speed uniformly chosen, and stays at destination for *pause_time*. Depending on the node's fixed parameter *stay_in_town*, it picks a new destination inside the town, goes there using the same method, stays at destination for another *pause_time*, and repeats the process, such that the total number of consecutive pauses inside the town is equal *stay_in_town*. The node then repeats the entire procedure from where it is now. In contrast, a *commuter* node represent mobiles that frequently commute from one town to another, having *stay_in_town* parameter equals to one and *pause_time* in a small value.

2.4.4.7 Obstacle mobility model

Authors in [61] extend their previous work [60] and proposed the design of a mobility and signal propagation model that can be used in simulations to produce realistic network scenarios (implemented in ns-2 [100] and GlomoSim [1, 139, 140] respectively). *Obstacle Model (OM)* allows the placement of obstacles that restrict movement and signal propagation, and also introduces a signal propagation model that emulates properties of fading in the presence of obstacles. Movement paths are constructed (based on *Voronoi Diagrams* [35]) as Voronoi tessellations with the corner points of these obstacles as Voronoi sites. As a result of [60, 61], a complete environment in which network protocols can be studied on the basis of numerous performance metrics were developed. The first component of *OM* is terrain modeling, where a user can define the positions, shapes and sizes of objects (buildings and other structures). The second component is a movement graph, which is a set of pathways along which the mobile nodes move, and is the *Voronoi diagram* [35] of the obstacle corners (see an example in Figure 2.32). The third component of the model is the route selection (*shortest path* routing policy issued to move between two locations, where the cost of each path segment is its Euclidean length). Finally, the fourth component is the signal propagation model.

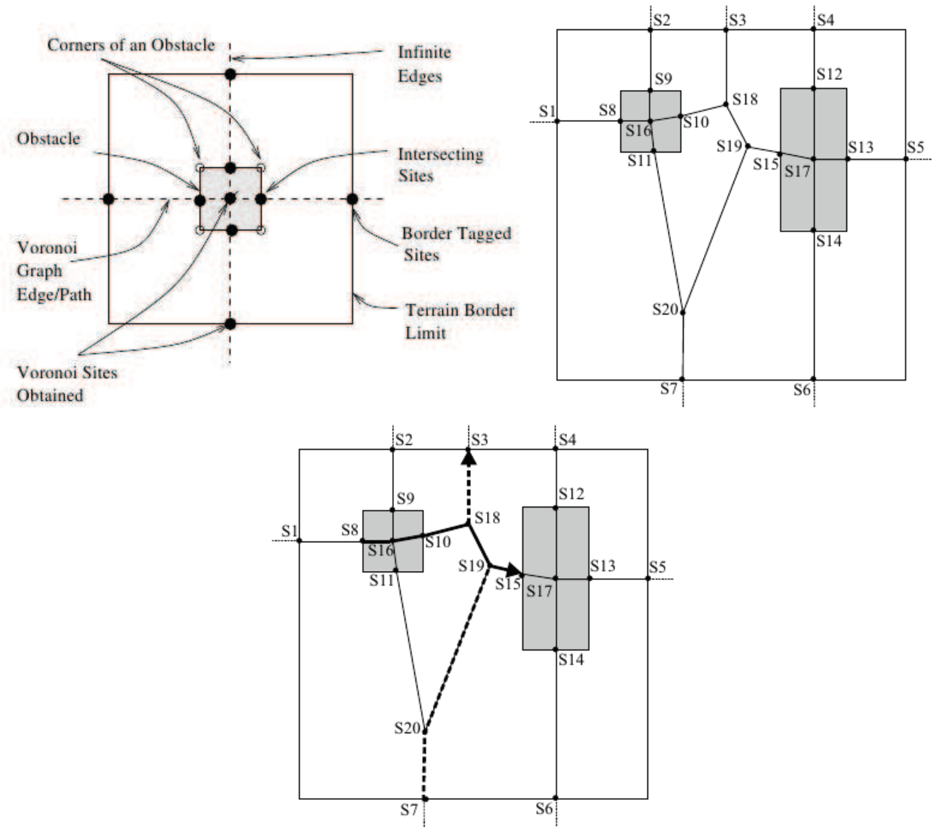


Figure 2.32: Examples of Voronoi graph, paths and movement (taken from [61]).

Object locations and connecting pathways are computed once at the beginning of the simulation and do not change during the course of the simulation. The initial placement of the mobile nodes is obtained by distributing the nodes at random locations along the pathways. Each node selects a destination location randomly and then moves to that location using the shortest route (shortest path on the graph created by the pathways) from its current location. Upon reaching its destination, the node pauses for some rest period. This process is repeated again until the end of simulation.

Some enhancements of the model included in [61] are the addition of *attraction points* and the use of a *weighted exponential distribution* for destination selection, based on real-world behavior, where (i) multiple nodes may move towards a certain place of interest at the same time or within a specific interval of time, and (ii) a person is more likely to select a destination that is close to his/her current location, and far locations are still possible, but with a small probability. Destination location is now chosen from a *weighted exponential distribution*, in which if m is the x^{th} farthest point from the current location of the node, the probability of choose m as destination point is given by $f(x) = ae^{-ax}$, where a is a constant that can be selected to modify the distribution.

The previous work [61] assumed that objects are substantial enough to prevent the passage of transmissions through their walls, and that nodes have omni-directional antennas. Thus, the model use *obstruction cones* (see Figure 2.33), and a *reachability matrix* to represent the reception likelihood of a transmission between a pair of nodes. Because there may be more than one obstacle in the omni-directional transmission range of a node, the node can have multiple obstruction cones. The *obstruction set* is then the set of all nodes located in the obstruction cones of a node. *Obstruction sets* can be represented as the following: $OS(node_i) = \{node_j \,|\, j\, is\, not\, in\, the\, line\, of\, sight\, (LOS)\, of\, i\}$, where $node_j$ represents a node j that lies in the obstruction cone of $node_i$.

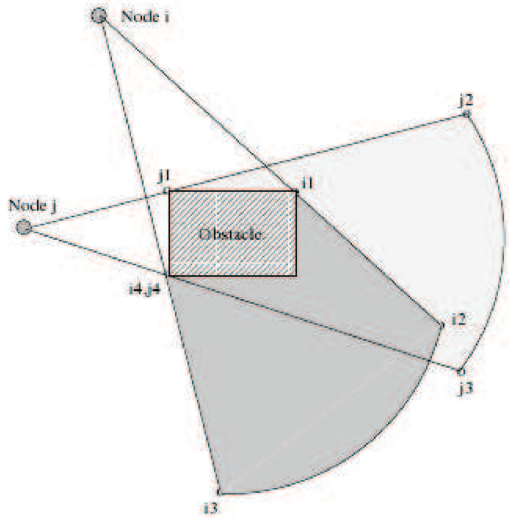


Figure 2.33: Obstruction cones example (taken from [60]).

The *Reachability matrix* is used to determine whether an object will influence the communication ability of a pair of nodes, and it is represented as shown in Figure 2.34. In this matrix there are four possible cases for the node-pair configuration, where a value of '1' indicates complete reachability, and '0' indicates a completely blocked transmission. As summary, if two nodes are within transmission range of each other, the possible cases are:

- **I-to-I**: if both objects are within the same obstacle and there is a straight line that can be drawn connecting the two nodes, the nodes are capable of communication with each other.
- **E-to-I** and **I-to-E**: A transmission between nodes from exterior to interior (or vice versa) is blocked because the nodes lie within the obstruction cone sets of each other.
- **E-to-E**: This mode of transmission is dictated by the obstruction cone of the source node. The packets are dropped according to whether the destination is a member of the source node *obstruction set*.

As the model of propagation and transmission described before is very restrictive and unreal, the *OM* authors proposed a new model, in which two nodes in a non-line-of-sight are able to

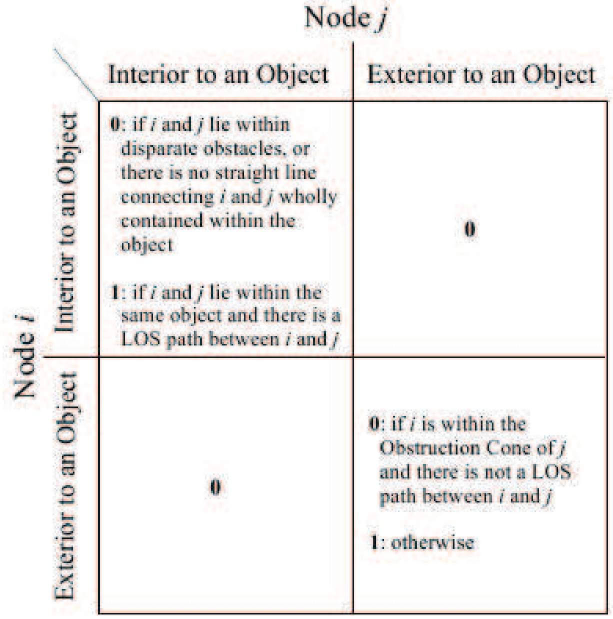


Figure 2.34: Reachability Matrix for a pair of nodes(taken from [60]).

communicate between them. *OM* use either the *Two-Ray Pathloss Model* that accommodates the reflections of the signals off the surface of the ground, in addition to the direct path signals from the source transceiver to the destination transceiver, or the *Friis' Free Space Equation* [114] which considers only a single path of propagation. At the same time, *OM* uses empirical values to model when signal cross-over an obstacle, defining some attenuation values (single or double wall), thus reducing the signal strength received.

2.4.4.8 Graph-Based mobility model

In [127] a mobility model is proposed, in which an area is represented as a connected graph, and places that nodes might visit are *vertices* and the connections between those places are *edges*. Each node is placed randomly in a vertex and selects a destination place randomly. After that, the node selects the shortest path between them, and moves on until reaching the destination place, where it will be stopped for a random pause time. Then, this process is repeated again.

Tian et al [127] include some metrics in order to characterize the coverage area, based on three concepts: (i) *the gross area* is the smallest rectangle that contains all *vertices* an *edges* of the graph, (ii) the *gross length* is the length of *gross area*, and (iii) the *gross width* is the width of the *gross area*. Those metrics are defined to compare *graph-walks* and *random-walks* (using the *random walk model*): *maximum radio coverage of graph-walk* ($CMax_g$), *maximum radio coverage of random-walk* ($CMax_r$), *radio coverage density* (D_g and D_r), and the *ratio between densities* (α).

2.4.4.9 Area Graph-based mobility model

This model is introduced in [16] and uses a directed and weighted graph as a boundary for the motion of the network nodes. Because real scenarios consists of several clusters (with high density) and fixed paths (with low density), the graph is modeled by several rectangular planes (*vertices*) and direct connections (*edges*) between them. The weight of an edge is the probability of a node choosing this edge when leaving the vertex. Waiting time in every vertex is chosen uniformly distributed from a user defined interval. The motion in the *Area Graph-based Mobility Model* consists of two parts: Motion inside vertices (determined by *RWP*) and motion between vertices. When a node enters a vertex, a waiting time inside is determined. When the waiting time is finished, an outgoing edge is chosen randomly (considering weights of the edges). Then, the node moves to the connection point of the vertex and the edge and thereafter moves with a randomly determined speed to the chosen vertex. An example of a topology is shown in Figure 2.35.

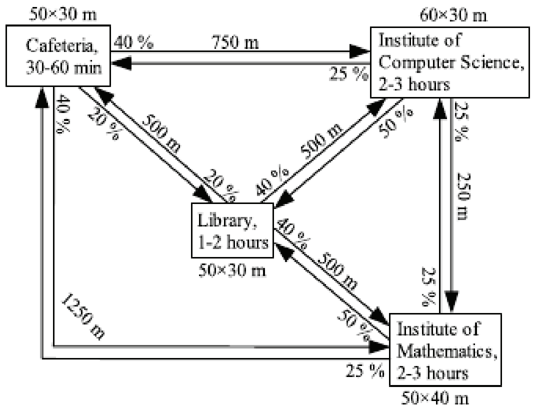


Figure 2.35: Area Graph-Based scenario example (taken from [16]).

2.4.5 Hybrid characteristics

The models described in this section refer to those models that combine different characteristics or more than one of the previous models.

2.4.5.1 Working Day Movement model

Working Day Movement Model (WDM) [37] was built as an extension to the *Opportunistic Network Environment (ONE)* simulator [65] and it was defined taking into account previous works for inter-contact time and contact time distributions. At the same time, authors of *WDM* highlight three conclusions from [54]: (i) nodes are very often turned on/off and visit a small portion of the area, (ii) mobility while using the network is very low, and (iii) repetitive patterns with period of one day and heterogeneity among nodes are revealed.

Intuitively, *WDM* models the behavior of people in a working day, going to work, having activities after work, and returning to their home, while people have three ways for moving, by walk like pedestrians, by urban transport like bus and by their particular vehicles like cars. A more detailed description of the model is as follows:

- There are three activities (being at home, working, and some evening activity with friends) and they are called *activities sub-models*, and differ one from the others.
- Sub-models repeat every day, resulting in periodic repetitive movement.
- Communities and social relationships are formed when a set of nodes are doing the same activity in the same location.
- Each node should start from home when its assigned wakeup time is reached (for one node, wakeup time is the same for the whole simulation). When nodes leave their homes, they could choose from using different transport methods or sub-models (by car or by bus) to travel to work. After the working hours, the nodes decide, by drawing, whether they go out for the evening activity, or return home. Different user groups have different locations where the activities take place.

As mentioned before, WDM consists in six sub-models and one map, and they are briefly described next:

- *Home activity sub-model*: nodes stay at home until wakeup time without moving.
- *Office activity sub-model*: a 2-dimensional model is used for movements inside the office, where each node has its desktop. The office is entered from a specific map point, and it is a square where the upper left hand corner is the door. Each node is assigned a coordinate inside the building where the node's desk is located. The movement inside the office is: the node starts walking towards the desk, and when it reaches its desk, it stops for an amount of time. When the node wakes up from the pause, it selects a new random coordinate, walks there and waits for an amount of time, and so on until work day is over.
- *Evening activity sub-model*: models the activities that nodes can do in the evening, i.e. after work, as groups. Each node is in the beginning of the simulation assigned a favorite meeting spot. Immediately when a node ends its working day, it is assigned to a group based on its favorite meeting spot. The node then uses the transport sub-model to move to the meeting spot. The node waits at the meeting spot until all the nodes of the group are present, then they start moving according to the map based movement model (random walk) on streets in a group for a certain distance, and then they pause for a longer time, and finally split up and walk back to their homes.
- *Transport sub-model*: consists in three sub-models, walking, car and bus. Nodes that walk use streets to advance with a constant speed towards the destination. Dijkstra's algorithm is used for looking the shortest path to the destination. Nodes owning a car can travel at a higher speed between different locations. Nodes without a car can use buses for traveling faster. There are predefined bus routes on the city map. The buses run these routes according to a schedule. Buses can carry more than one node at a time.
- All nodes move on a map, which define the space and routes in which the nodes can move; the map contains all the information of the locations of the houses, offices and meeting spots, as well as the bus routes with bus stops.

The principal drawback is the complexity of configuration due to the high level of detailed information modeled. Configuration of homes, activities, bus routes, etc., is a time consuming task, valid only for one scenario. Even more, there should be some real measurements that validate them, in order to get more accurate results. Floors, walls and different obstacles are not covered, and activities could be more detailed (e.g., lunch breaks and shopping activities) or refining some mobility model used in activities. Furthermore, devices are always turned on, which is not always true for laptops.

2.4.5.2 Disaster Area model

The *Disaster Area* [4] model movements of nodes as they were in a catastrophic situation (see Figure 2.36), in which nodes exhibit structured movements based on the civil care and protection. Its main characteristics are: heterogeneous area-based movement, obstacles, and joining/leaving of nodes.

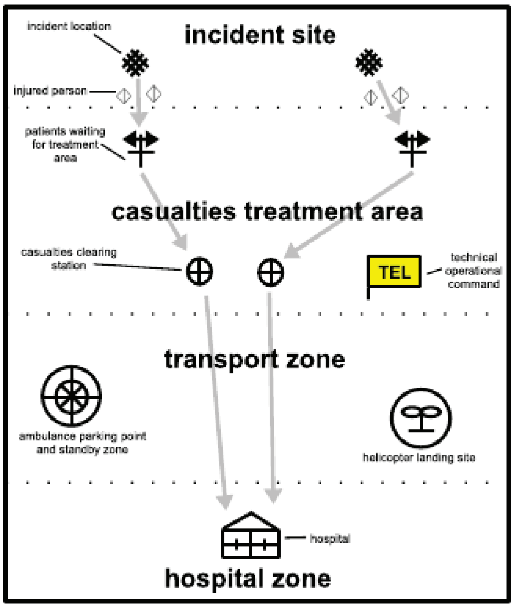
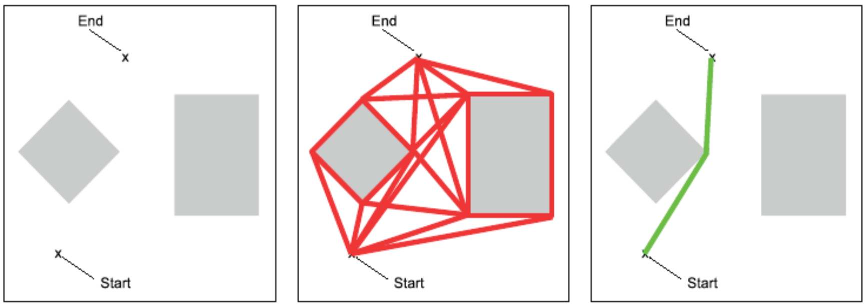


Figure 2.36: Disaster area scenario overview (taken from [4]).

To implement the area-based movement, the simulation area is divided into disjunct tactical subareas. These areas are classified in *incident location (IL)*, *patients waiting for treatment area (PWT)*, *casualties clearing station (CCS)*, *ambulance parking point (APP)*, and *technical operational command (TOC)*. Technically, a disaster area scenario S consists of a simulation area F, a set of tactical (sub-)areas R, and a set of obstacles H. For each tactical sub-area, two sets of nodes are defined (stationary and transport nodes), where nodes in the set first only move inside the sub-area and nodes in the second set can move to the next sub-area to carry patients. Is important to note that pedestrians and transport nodes have their own velocities intervals, which differs between groups. The movement of transport units depends on the class of the tactical area the node is assigned to, being the most interesting case when it is an *Ambulance parking point*. In this case, after it moves to a randomly point chosen inside *casualties clearing station*, then wait for some time and leave the scenario.

The optimal path for the movement of the transport units between the different areas and avoiding obstacles is determined by robot motion planning methods [35], using visibility graphs. A visibility graph is a graph where its vertices are the vertices of the obstacles (and the entry and exit points of the areas), and there is an edge between two vertices if the vertices can *"see"* each other, meaning the edge does not intersect the interior of any other obstacle. Thus, after calculating the visibility graph, the shortest movement path between two areas for each transport unit can be calculated using the weight of edges (Euclidean distance) for the Dijkstra's algorithm. Figure 2.37 shows an example: (a) a map with two obstacles and a start and end point, (b) the visibility graph, and (c) shows the resulting shortest path between the start and end point.



(a) map (b) visibility graph (c) optimal path

Figure 2.37: Finding optimal path (taken from [4]).

2.4.5.3 User based models

In approaches that follow a user-oriented strategy [123, 124] movements are governed by social behavior. This kind of work relies on some part on the use of maps and points of interest, and this is not a desirable properties for the research reported in this thesis, because it would be necessary to define maps manually for each simulation. The other restriction with this is that paths taken by nodes to go to the next place in its itinerary are generated using Dijkstra shortest-path algorithm [36] or some other method that do not models reality for children of *Plan Ceibal* networks.

Several studies have been directed by applying social network theory, like in [39, 91, 92, 93, 134].

2.4.6 Trace-based mobility models

In this section are described models that are generated based on a set or real movement traces.

2.4.6.1 Mobility model from a heterogeneous military MANET trace

In [80] a mobility model is proposed, based on analysis of a real trace collected from a military experiment. The structure of these entities in the trace is novel, because they are layered and heterogeneous: some nodes move on the ground whilst some are hovered in the sky, like jeep vehicles and Unmanned Aerial Vehicles (UAVs) as shown in Figure 2.38.

Traces logs have every vehicle's ID, *GPS* location and communicational path-loss data throughout the period per second time. For this model, two nodes are out of communication when the pathloss between the two nodes is higher than a threshold. A node is said to belong to a specific group when it is within the communication range of any member in that group. Nodes of the same group always move as a unit group, in the same direction, with relatively the same speed, and exhibit identical mobility behavior. For modeling changes in trajectories a distribution of absolute relative direction angle is derived (the angle in which a vehicle changes its direction), concluding

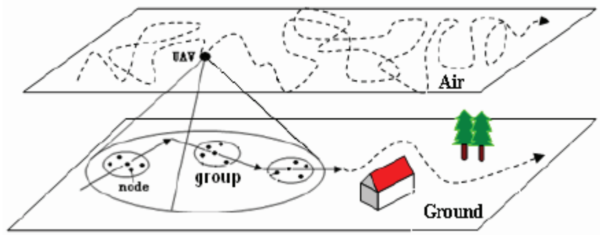


Figure 2.38: Heterogeneous and layered model (taken from [80]).

that a mobile node does not select its new direction randomly from a uniform distribution. In the same way, travel duration and pause duration are derived from traces.

Finally, rule sets defining the mobility model are presented at next.
Structural rules set:

- nodes are organized into groups and nodes within a group have the same travel duration and pause duration.
- Within a group, nodes can be heterogeneous and have different mobility flexibility
- Each group has a FIFO destination queue to save the location of destinations.
- Groups start off at different times and have heterogeneous travel schedules.
- Nodes of a group start at the same place in the beginning.

Mobility rules for a group:

- Step 1: All nodes of a group pause for a certain time, given by the pause duration.
- Step 2: After the pause duration, if the destination queue is empty, go to **Step 3**, otherwise go to **Step 5**.
- Step 3: Generate a group destination with relative direction angle. Broadcast this new destination and put into all group's destination queues.
- Step 4: Calculate the group's travel duration according to the distance to this destination and the average group velocity.
- Step 5: Pop a destination from the group's queue. Every node of this group calculates its destination around the group destination randomly in a circle region whose radius depends on the type of node. Heterogeneous nodes have different mobility radius (see Figure 2.39).

Step 6: Every node of this group starts off to its destination.

2.4.6.2 Urban Pedestrians Flow

Because of costs, authors of *Urban Pedestrians Flow (UPF)* identified the need for a method to reproduce pedestrian flows from simple observation such as density observation on streets, which can be taken by fixed point observations using web cameras, volunteers and so on. *UPF* [83] focuses on the behavior of pedestrians in urban areas and it proposes a new method to generate a mobility scenario that classify pedestrians into multiple groups by their similar behavior patterns

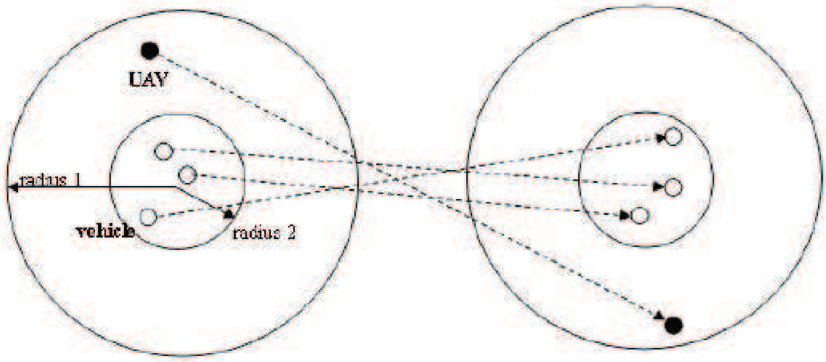


Figure 2.39: Movements of different type of nodes (taken from [80]).

(flows). Given the observed road density in the target field, *UPF* derive how many pedestrians per minute follow each flow by using *linear programming* techniques (minimizing the maximum error between the road density observed and the road density derived). Using the derived flows, it is possible to generate a *UPF* scenario which can be used in network simulators, in particular in the *MobiREAL* simulator. One interesting feature of this simulator is that it can generate/delete mobile nodes according to the *UPF* scenario.

To generate urban pedestrian flows, a *scenario* is given as input, modeled as a graph where corners and streets are the nodes and edges of it. This edges and nodes are between polygons, that represent obstacles (Figure 2.40). Another input are *Pedestrian Behavior Patterns*. People are classified in groups, and for each group, an estimation of their behavior patterns as a route (this is called a flow) is made. Finally, the averaged observed density of pedestrians in some streets of the field are needed, in order to compare accuracy with generated flows.

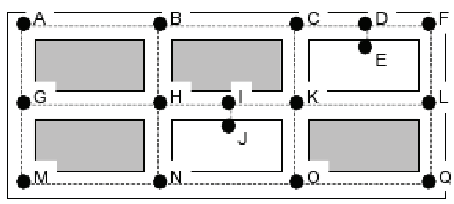


Figure 2.40: Simulation field graph (taken from [83]).

2.4.6.3 Condition Probability Event

Condition Probability Event (CPE) [67] is a model to describe the behavior of a node that changes its route or speed dynamically according to the situation. The behavior is described as a list of rules where each rule consists of a *condition*, a *probability* and an *action*, and internal/external variables exist. The external variables (simulation clock T, surroundings information of the node E, output data from the network system to the node AO, input data of the node to the network system AI, current position P and velocity vector V of the node) can be accessed (and updated) from the outside of the *CPE* model. A logical formula using the variables can be specified as a *condition*, and a value [0,1] or a probability function can be specified as a probability.

As an *action*, a set of substitution statements which update values of the variables can be specified.

The model is executed as follows. The simulation clock T is incremented automatically, and for each increment, an executable rule is searched from the list, and a rule that satisfies its condition and probability is selected to be executed. This search ends if it executes a rule, or when reaches the bottom of the list of rules.

2.4.6.4 Model T and model T++

In [76] the authors present an empirical (build from real traces) registration model (*Model T++*) derived from the WLAN registration patterns of the mobile users, and the main contribution is that the model is able to formulate the inter-dependence of space and time explicitly by a set of few equations. This work is an extension of a previous work [59] (*Model T*), which adds a joint time-space dependencies to the registration model. The focus in *Model T++* is to capture the fact that a simple but proper notion of popularity gradient is enough to capture the correlation between space and time. Indeed, when locations (i.e., *AP coverage area*) are differentiated with respect to the number of visits they are receiving (i.e., *AP popularity*), the time spent at each location i before user moves from i to k turns out to be closely related to the difference of popularity between locations i and k. Other related works [66, 130] exists, and all of them study the user registration patterns of various campus *WiFi LAN* data, extract useful features that are representative of the complex patterns of how users change their associations across *APs*, and abstract a mobility model at varying granularity, but none of them have time-space correlation.

Model T++ accepts the number of users and the number of locations as input. The locations are then partitioned into disjoint clusters, where the number of clusters and cluster sizes are either provided as inputs, or otherwise the model uses the cluster information from the *Dartmouth* data [68]. Both the Intra-cluster and inter-cluster transition probability matrices are generated from the parameterized model using *Weibull* distributions. Each user makes a number of transitions inside its current cluster before making a transition to a new cluster. The transition decisions between two locations inside the same cluster, as well as between the two clusters are performed according to the transition probability matrix.

The process of synthetic trace generation starts with the number of mobile users n, and the total number of *APs*. A spatial trace consists of a sequence of *AP IDs* (including the *OFF* special state), and the initial placement of users at *APs* is assumed to be random. A complete synthetic trace for a user is generated by starting from a random cluster, and making a transition to one of the other clusters as prescribed by the inter-transition model. Once a destination cluster is selected, a number of intra-transitions are made in that cluster before moving to another cluster. The number of intra-transitions is determined by the intra-cluster length model. When an inter-cluster move is made, the process is repeated again until a number of inter-cluster moves is completed. The number of inter-cluster moves is also determined by the inter-cluster trace length model. The full trace generation process is repeated for all users.

Even when Model *T++* is an elegant way to denote mobility without complexity of obstacles, the user registration patterns corresponds to a dense campus *WiFi* network, so it is not correct to directly applicable on general mobile ad hoc networks. Model *T++* has vulnerabilities if an actual geographical mobility where the positions of the users are required for the network simulation.

2.4.6.5 Evolutionary topology mobility model

In [86], authors analyze the mobility patterns of wireless hand held *PDAs* users in a campus, and characterize the high-level mobility and access patterns to compare them against studies focused on laptop users. Additionally, authors develop two models: *Evolutionary topology model* and *Campus Waypoint model* (see Section 2.4.6.6). The *evolutionary topology model* is a constructive model based upon the mobility of the users as well as the wireless connectivity. A compelling feature of the model is that it incorporates the wireless connectivity and propagation characteristics. Consequently, it naturally captures and models the range, interference, and obstruction properties that are challenging to realistically model using analytic approaches. Intuitively, this model represents connectivity among users based on network proximity, i.e. if two users can reasonably "hear" each other.

The characterization of overall user mobility has two perspectives:

- the distribution of the number of access points with which users associate and the number which they detect (i.e. how widely users in the trace roam across campus while using their *PDAs*).
- the distribution of the number of users which associate with particular access points (i.e. how concentrated this roaming is). Authors find that students are relatively mobile and use their *PDAs* in many locations.

An interesting aspect is that compared to the laptop users in the Dartmouth study [68], authors found that the typical wireless *PDA* user is over twice as mobile as the typical laptop user in terms of associated access points. This indicates that *PDA* users tend to operate in a larger number of locations than their laptop counterparts.

For each time slot, a node is created in the topology for each active user in the trace, and nodes are connected by edges if users' wireless devices could reasonably communicate with each other at that location and time. The connectivity between two users is approximated by creating an edge between two nodes if the intersection of the set of *APs* sensed by their *PDAs* is non-empty, and the edge is removed if the intersection of *APs* becomes empty again. Nodes and edges appear and disappear based upon *PDA* on/off events and user movements observed.

The evolutionary model emphasizes the limitation of the popular *two-ray ground reflection model* for radio propagation in an environment containing obstacles. Authors show some topology examples in which topology created by evolutionary model are close to reality, but the *two-ray ground model* would create a completely connected graph under typical radio settings.

2.4.6.6 Campus Waypoint mobility model

Campus Waypoint model (CWM) is one of the two models introduced in [86]. In this model, users are associated with geographic locations on campus, and model their mobility vectors and potential interactions as they access the wireless network over time. However, rather than choosing user locations, speeds, and directions using random distributions, access and mobility patterns of users derived from traces are used.

The campus waypoint model serves as a trace-based analog to the random waypoint model, but choosing user locations, speeds, and directions from access and mobility patterns of users in its trace. For each time step in the model, the user location is estimated via trilateration among the locations of sensed APs. *CWM* models the user mobility over time based upon: (1) the evolving set of sensed *APs*, and (2) the disappearance and reappearance of users at different *AP* locations on campus assuming reasonable velocities. The location of a continuously connected user over time determines the mobility rate, direction, and pause time of that user. The mobility patterns from these kinds of users generally represent roaming within a building or within building clusters. When a user disassociates from the network and re-associates at a different location, it generally represent outdoor roaming among buildings across campus. In this case, the direction is the vector between the locations of disassociation and re-association, and the speed is obtained by computing the geographic distance between locations and dividing by the time between associations.

The authors of *CWM* found that some mobility characteristics differ from those used in typical synthetic models:

- unlike in synthetic simulations, only a small percentage of users are actually in motion at any one time.
- users move in the *campus waypoint model* at an average speed of a meter per second, when the default node speed for ad-hoc routing in *ns2* wireless scenarios draws from a uniform distribution between 0–20 meters per second.
- users appear and disappear from the network. This behavior, absent in most documented simulations, can and certainly have drastic effects on network topology and connectivity.

2.4.6.7 SUMATRA

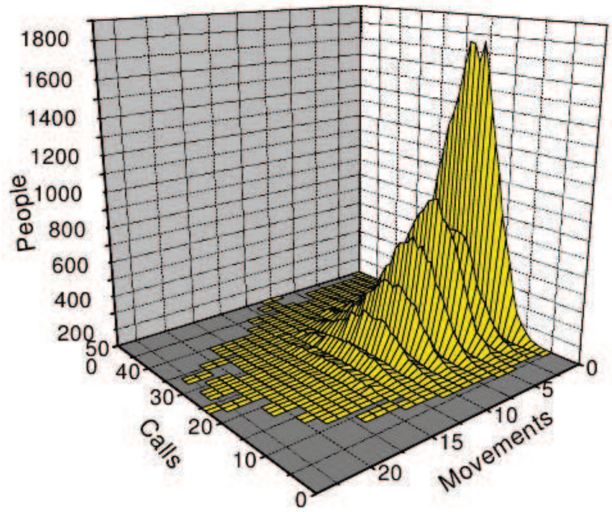
In [49] a summary of *SUMATRA (*Stanford University Mobile Activity Traces*)* is made, describing this trace generator, that was validated using real data for it (downloadable from [108]): *SULAWESI, S.U. Local Area Wireless Environment Signaling Information* and San Francisco Bay Area (*BALI, Bay Area Location Information*). The traces contain the following information for a call and move event:

- Call: the ID's of the caller and called mobile user, the zone ID's in which they are being, the time when the connection is started, and finally the duration of the call.
- Move: the ID of the mobile user, the current zone ID, the zone ID in which the user moves as next, and the time of the movement.

Unfortunately, as mentioned by the authors of *SUMATRA*, the *SUMATRA* traces have some disadvantages and thus it is difficult to use them for ad-hoc network simulations. The main problem is that it is not specified how a user travels from one zone to another. Since the velocity of the user is unknown, the model uses a global velocity of 15 mph defined. Therefore, you cannot figure out how long a travel lasts, which positions the user visits, and when it finally reaches the final position.

Anyway, the authors mark that they have obtained relevant data from the *BALI* real traces. Figure Figure 2.41 on page 46 shows a typical Gauss distribution and the movements show an exponential distribution (i.e. there are many people with little movements and few people with many movements); and Figure 2.42 shows that the number of people with odd movements are negligible. The vast majority of the people perform an even number of movements (the reason for this could be that the most people are commuters which do the same number of movements from home to work and back).

Figure 2.41: Histogram of movements and calls (taken from [49]).



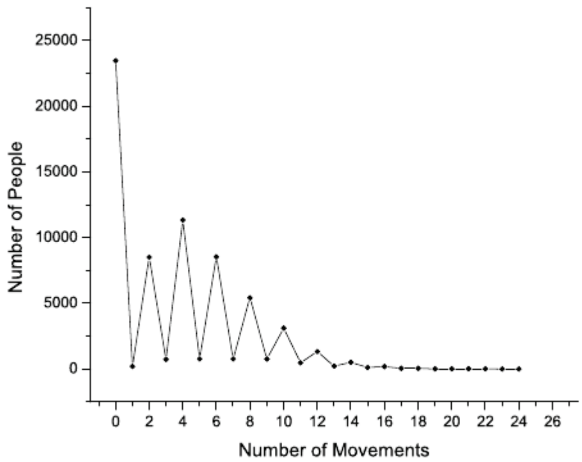


Figure 2.42: The number of people as a function of the number of movements (taken from [49]).

2.4.6.8 PCFG based mobility trace generation

In [42] a novel method is presented, based on *Probabilistic Context Free Grammar (PCFG)* [43], which is a generalization of *CFG* with probability values for each production rule. The approach is to infer a *PCFG* as output of the process, taking real traces as process input. In this way, *PCFG* is a concisely representation of movement patterns.

A mobility PCFG is defined as five-tuple $< S_{nt}, S_t, R_g, Prob, Start >$ where:

- $Start$ is the initial nonterminal symbol of the grammar,
- S_{nt} is a list of nonterminal symbols defined by production rules,

- S_t is a list of terminal symbols which are the symbols actually seen in the sentences,
- R_g is a list of production rules that map a string of terminal and nonterminal symbols onto a nonterminal symbol,
- $Prob$ is a list of probabilities, each one of them assigned to a rule to define the probability that this rule (as opposed to the other rules forming the same nonterminal) is chosen in parsing or string generation.

To capture node movements, a *PCFG* can be built using terminal symbols for both location and temporal information and derivable probabilities from reality or domain application. As an example, a sentence looks like $l_A\, t\, t\, l_B\, t\, l_C$, which means that a node stay in position l_A and two time units after it will be in position l_B, and finally, after one time unit in l_C.
For the automatic construction of a PCFG, an inference algorithm (see details in [43]) consist of: (i) data incorporation, and (ii) application of operators. Once the grammar is constructed, synthetic trace generation is performed, creating a sentence for a node describing its movements given a initial position. As interesting features of this approach remarked by authors are: *"is able to extend the pattern lengths according to the training data. Furthermore, the automatic construction method given provides generalization, hence unseen, but probable patterns are also added into mobility grammars of nodes"* [42].

2.4.7 Connectivity and opportunistic approach mobility models

This section describes the models created specifically for they use in opportunistic networks, or are models that consider only connectivity information, leaving in the background the need to locate nodes in real positions.

2.4.7.1 Connectivity Trace Generator

The increase of research in Opportunistic Networks or Delay Tolerant Networks paradigm raise the need to review in which traditional simulations are done. In a traditional way, all the pattern movement of nodes are described, but this approach leads to a long simulation time, high computing complexity and, in the best case, moderated fidelity. On the other hand, Delay Tolerant Networks are characterized by disconnectedness and sparse nodes, so in some cases it is possible to use connectivity models instead of mobility models, because *"what mobility models do is to provide a mechanism to generate permutations in the network connection graph"* [64]. An example of usage of this approach is [132], in which no mobility model is used, but connectivity graph and its changes instead, where the main aspects for modeling opportunistic networks are: *inter-contact times* and *contact durations* [28] and the behavior of cluster of nodes using frequency of connection events. An example of usage is described in [102].

The *Connectivity Trace Generator (CTG)* defined in [24, 25] is constructed in two phases: (i) determine connection parameters, and (ii) generate connectivity traces. Initially, the connectivity model is defined as $M = (K, P_R, P_C, N)$ where N is the total number if entities, K is a set of clusters $[K_1, ..., K_k]$, $P_R(\mathcal{A}, \mathcal{B})$ is a relationship function defining the probability that $\mathcal{A}$ and $\mathcal{B}$ are related, and $P_C(e)$ is the connectivity function defining the probability that two nodes related by e are currently connected. Here $e \in E$, the set of edges by applying P_R to all pairs of nodes.

The relationship model is ruled by two parameters, the intra-cluster relationship distribution λ and the inter-cluster relationship distribution ϕ, defining P_R in terms of them, thus:

$$P_R(\mathcal{A},\mathcal{B})\begin{cases}\lambda; & where\,\mathcal{A},\mathcal{B}\in K_i\\ \phi; & where\,\mathcal{A}\in K_i,\mathcal{B}\in K_j\,and\,i\neq j\end{cases}$$

2.4.7.2 GeSoMo

GeSoMo [38] stands for *General Social Mobility*, and is designed specifically for modeling in *Delay Tolerant Networks*. Social Mobility Model models the social aspects of human mobility, and each one needs a model of the relations between a set of relevant people (called *Social Network Model – SNM*) in order to simulate their mobility. Previous related woks in this area [19, 20, 40, 57, 92] are simplistic and make a bundle between *Social Mobility Model (SMM)* and the *SNM*, so is not possible to change *SNM*. *GeSoMo* is a *SMM* that generalizes a number of existing models, separating the *SNM* of the *SMM*, and being able to choose what SNM to use. This is a very important feature, because some properties of social networks are dependent of the scenario [2]. The motivation of GeSoMo is based on the fact that there are a few human mobility characteristics that should be captured in an accurate model, but actual models [19, 20, 40, 51, 57, 92] do not have them all together:

- *Inter-contact times:* the aggregated inter-contact CCDF is characterized by a power-law up to an inter-contact time of about half a day, thereafter followed by an exponential decay, independent of the investigated trace [29, 63].
- *Temporal regularity*: the probability that individuals return to previously visited locations after a certain time t is characterized by different peaks at multiples of a day [46, 55].
- *Spatial regularity*: individuals have a strong preference for a small number of locations while visiting all other locations only with low probability [46, 55, 130].
- *Group mobility:* is a realistic human characteristic.

An overview of GeSoMo and its related concepts is shown in Figure 2.43 on page 49. At the top of the diagram, *SNM* appears as the generator of the input social network. *SNM* does not cares about mobility itself, but models social interactions instead. Previous research has found that some properties characterize *SNMs*, like strong clustering [131], small average path length [88], power-law node degree distribution [8], hierarchical communities [141] and assortative mixing [98]. Social interactions are formally defined as follow: A social network is a weighted undirected graph $G = (V, E, w)$ where $V = \{v_1, ..., v_n\}$ is the set of nodes, $E \subseteq \{\{x, y\}|\, x, y \in V\}$ defines the set of social relations with $\{u, v\} \in E$ if nodes u and v share a social relation (*social acquaintances*), and $w^{u,v} \in [0, 1]$ represents the strength of the social relation $\{u, v\} \in E$, with $w^{x,y} = 0$ if $\{x, y\} \notin E$.

Once the Input social network is generated, SMM generates mobility traces from it, by translating the static structure of a social network into a spatial-temporal structure of a mobility trace. After that, a simulation with this mobility trace is made, and a connectivity graph is constructed, in order to be used to maintain conformance between the connectivity graph and the input social network, based on a *conformance* metric. Periodically, a conformance test is performed, in which the number of meetings between two nodes must be proportional to the weight of the social relation in the input social network, and some actions are done depending on the result of it (for example, isolation, in which a node is isolated in order to get down it number of meetings). The elements in *GeSoMo* are *mobile nodes* and *anchors* (a place where a social interaction takes place). A mobile node moves between anchors, and when it reaches an anchor, stays there for a specific duration, and then starts to move again to the next anchor. Movement of nodes is controlled by both *anchor* and *node* attractions and repulse factors.

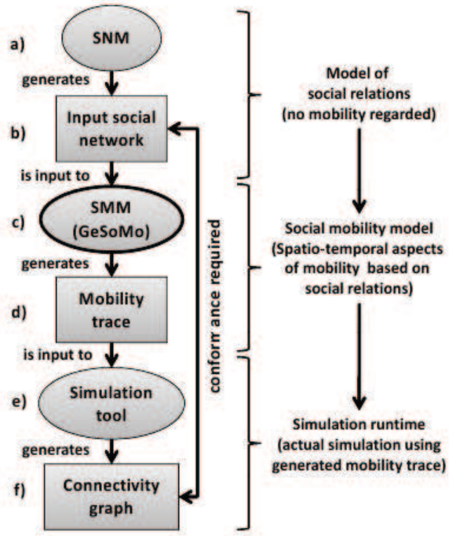


Figure 2.43: GeSoMo conceptual view and related concepts (taken from [38]).

2.4.8 Frameworks

This section describes the related works that are not only a mobility model, but a complete framework instead. These works are a more general tools to generate mobility, taken real traces or connectivity parameters as input.

2.4.8.1 TVC model

Authors of the *TVC Model* [58] remark that a good mobility model should capture *realistic* mobility patterns of scenarios, being *mathematically tractable* and being *flexible* enough to provide qualitatively and quantitatively different mobility characteristics by changing some parameters of the model, yet in a repeatable and scalable manner. An important property of this model is that is a *synthetic mobility model* that captures non-homogeneous behavior in both space and time, and at the same time, match two prominent properties, *location visiting preferences* and *periodical re-appearance*, that seems to govern human activities [55].

Another novel characteristic of the *TVC Model* is the optional feature of adjust the user *on-off pattern*. In many scenarios, nodes are not always-on, for example, in a *WLAN* or in a *DTN* nodes (e.g. laptops) are often turned off while people moves from one location to another, and the "off" time is often not negligible [55]. *"Finally, in addition to the improved realism, the TVC model can be mathematically treated to derive analytical expressions for important quantities of interest, such as the average node degree, the hitting time and the meeting time"* [58].

This model was validated against traces from multiple scenarios, which is an important feature in order to make this model a general one for different scenarios like *WLAN* traces, human and vehicular networks.

2.4.8.2 MobiREAL

The *MobiREAL* network simulator [67, 84] was developed to reproduce realistic environment of any city section based on two mobility models, *Urban Pedestrian Flows (UPF)* [83] which focuses on the densities and flows of pedestrians in city section, and *Condition Probability Event (CPE)* [67] which allows nodes to change their behavior dynamically according to the context of network applications. In order to make it easy to generate mobility scenarios, a support tool in *MobiREAL* is provided by the authors, together with a visualization tool named *MobiREAL Animator*.

2.4.8.3 SMM generator

In [137] a trace-driven framework capable of building realistic mobility models is proposed, by combining coarse-grained wireless traces with a map of the space over which the traces were collected. Through a sequence of data processing steps, a probabilistic mobility model that is representative of real movement is generated (see Figure 2.44). *SMM* provides a midpoint between purely hypothetical models and fine-grained observations, generating plausible models that adequately match real behavior, without significant costs or infrastructure beyond common wireless access deployments.

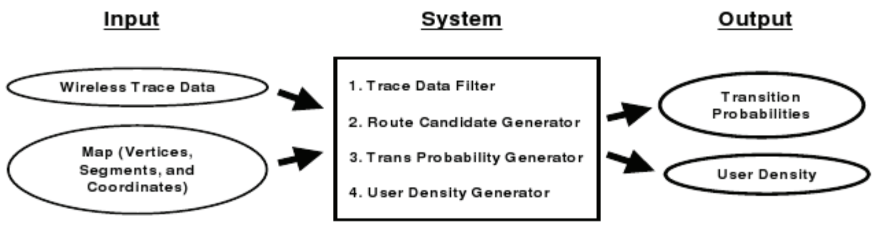


Figure 2.44: SMM overview (taken from [137]).

Intuitively, the model works as follow: (i) initially it is populated by estimating user densities at each map location (from the data trace), (ii) with the map and a set of heuristics, the user movement is modeled as a second-order Markov chain and the model generates from the filtered data trace a set of transition probabilities from one map location to another. For the generation, the authors used a defined metric called *average stay time*, in order to recognize *trips* inside the real traces. In the process of routes calculation, they used the *distance* as the only metric, and the model searches for N shortest paths instead of only one. By using information obtained from routes distances, the *transition probabilities* are derived, based on a proportional value in which shortest routes are more frequent. Finally, a Markov chain is used, in conjunction with transition probabilities in the following way (an example is shown in Figure 2.45 on page 51) $Prob[next \mid current, previous, origin, destination]$.

2.4.8.4 MOMOSE

Authors in [21] defined *MOMOSE* as a mobility model simulation tool for *MANETs*, which is extensible and has high adaptability to the information needed in order to evaluate a specific protocol. The set of nodes is partitioned in subsets, where each subset has its own mobility model during all the simulation. *MOMOSE* allows the simulation of obstacles too, making movement more real, and reflecting changes in the attenuation of transmission signals. This software is publicly available in [104].

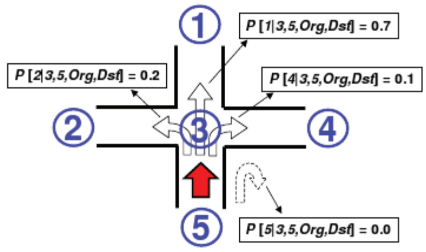


Figure 2.45: Example of usage of transition probabilities (taken from [137]).

2.4.8.5 CosMos

CosMos [49] stands for *Communication Scenario and Mobility Scenario Generator for Mobile Ad-hoc Networks*, and aims at the modeling realistic scenarios, where nodes changes its mobility characteristics (i.e. the mobility model and the velocity) several times. Approach taken by *CosMos* combines a wide variety of well understood random mobility models with a graph-based zone model, where each zone has its own mobility model and parameters. The scenario is a combination of directed, weighted graphs where the weights correspond to the flow of mobile nodes between neighboring zones and zones with different mobility models. A zone characterizes a certain geographical area and has several properties (*mobility model, population, neighborhood, and geometric shape*), and nodes move on it according to those properties. *Neighborhood* property defines the neighbors of the zone, and the *exit probability* of a zone specifies the rate with which mobile nodes move to neighboring zones. Thus, the graph is composed by nodes (*zones*), and the *neighborhood properties* define weighted and directed edges as shown in Figure 2.46 on page 52.

The movement of nodes in *CosMos* is similar to the described in [16], and it presents two types: *intra-zone* and *inter-zone* movements. Here, *inter-zone* movements are modeled as a Markov-Chain. The state is given by the zone number in which the mobile node is being. The transition matrix M is derived from the *exit probabilities* of the zones. The state probability for a given step j is given by $p_j = p_0 \cdot M^j$. Furthermore, the steady state distribution of the nodes, which is independent from the initial distribution, is given by the equation $\pi = \pi \cdot M$. Let $n = n_1 + \ldots + n_k$, where k is the number of zones, be the total number of mobile nodes in the simulation world. Since all nodes behave independently, the average number of nodes n_i in zone i is given by $n_i = n \cdot \pi_i$.

The authors of the *CosMos* framework remark some extra features of the tool like: (i) it permits to design complex simulation scenarios (ii) it provides a set of mobility models, and a simple interface extend available mobility models, (iii) it provides a set of predefined communication models, (iv) the tool generated mobility and communication patterns are can be used with the *ns2* network simulator, but can be easily extended to support other simulation tools, and finally (v) it can generate files to enable the researcher to check the simulation scenario before starting long simulations.

2.4.8.6 MoNoTrac

The *Mobile Node Tracer (MoNoTrac)* is introduced in [48] and is a tool to generate mobility traces based on geographical (and real) data provided by the *OpenStreetMap* project [106] with a plug-in interface to allow customization and adaptation (see Figure 2.48 for a system overview).

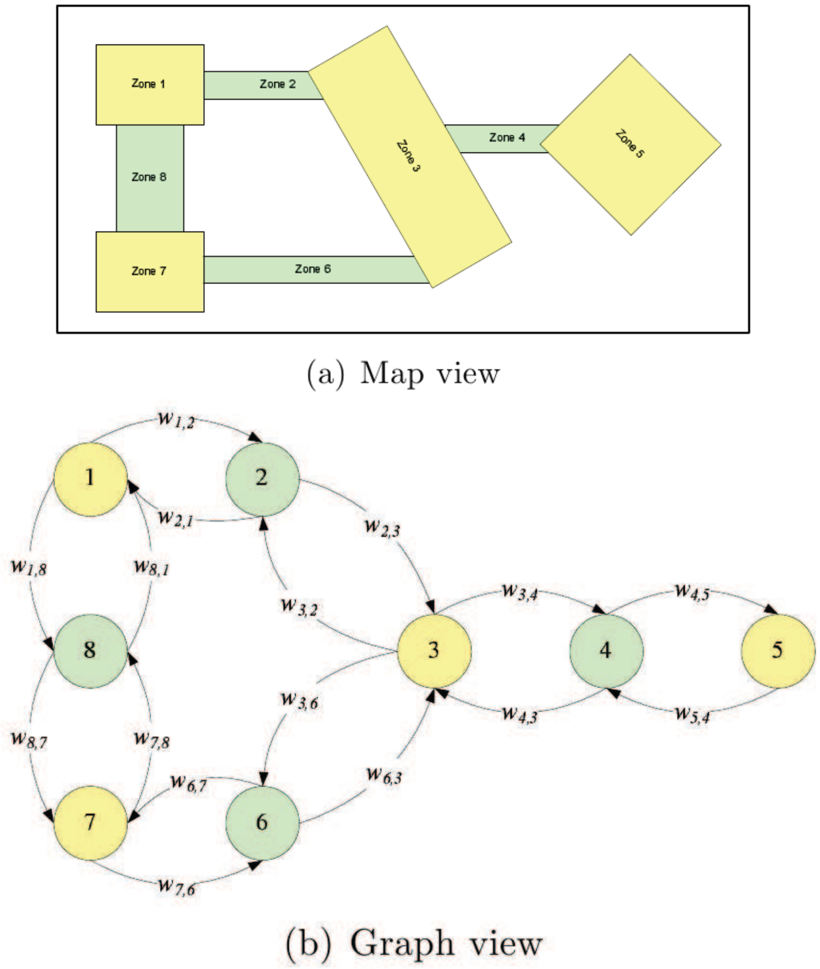


Figure 2.46: Example of representation of real scenario (taken from [49]).

MoNoTrac takes as input scenario descriptions, which consist of a movement area, mobile nodes, mobility models, and a simulation time to generate mobility traces. Users create a scenario by selecting a region with streets, roads, and public transportation stations from a repository of geographic data that serves as movement area for mobile nodes. After that, the number and type of mobile nodes are added to the scenario description, and for each node, a mobility model is applied to them. One interesting feature is that the movement area can be bounded or boundless. Thus, nodes can leave, enter, and reenter the simulation area if required. The number, type, and distribution of mobile nodes can be specified. Currently, the types *pedestrian* and *car* are supported. Even when this work, and some proposed extensions have important features, it is still a work-in-progress [103].

2.4.8.7 IMPORTANT

IMPORTANT [6] is a framework, and it stands for and was designed for systematically analyze the *Impact of Mobility on Performance Of RouTing protocols for Adhoc NeTworks*.

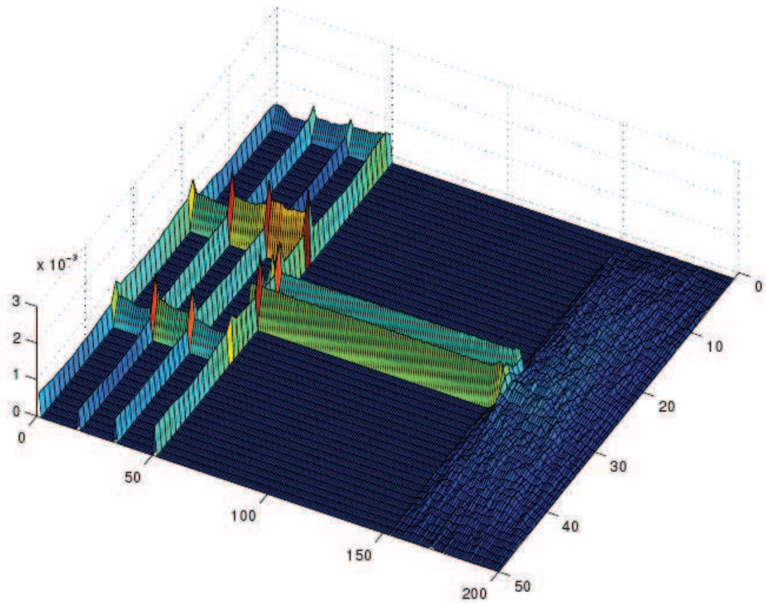


Figure 2.47: Example of simulation scenario (taken from [49]).

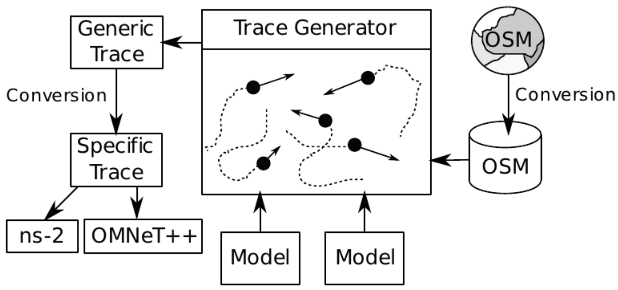


Figure 2.48: System Model (taken from [48]).

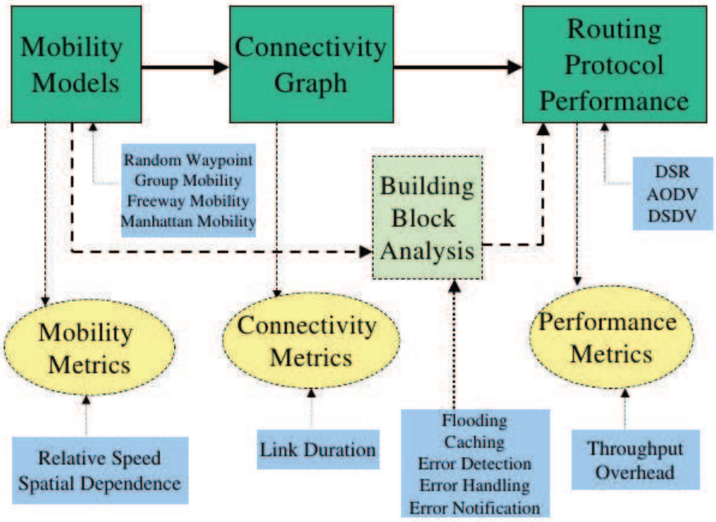


Figure 2.49: IMPORTANT framework (taken from [6]).

The main goal of this framework is to analyze the behavior of *MANETs* under different mobility models (and routing protocols), and this is the reason why authors include two existent models (*RWP* and *RPGM*) and also define two new more (*Manhattan* and *Freeway* mobility models). In this sense, the most relevant contribution is to give a framework to systematically get results of mobility metrics for each model, in order to compare and conclude about characteristics that affect to routing protocols and applications.

2.4.8.8 Connectivity Trace Generator

Connectivity Trace Generator (CTG) [25] is a tool for testing opportunistic mobile systems, and it uses the connectivity model defined in [24]. This tool is capable of generate different traces (for example varying number of hosts) having similar connectivity patterns as real ones, but obtained from only one set of real traces. As authors summarizes: *"from real traces, generates synthetic realistic traces"* [25].

The key aspects of *CTG* are represented in Figure 2.50 on page 55. First of all, an empirical derivation of connectivity distributions from real traces is made to generate parameters needed by the tool. Parameters derived by authors are: the *co-location* of two users (as a function of the probability for a user being in a specific place for a given time), the *residence time* and the *degree distribution* of the nodes. At the second stage, the trace generator allows the generation of synthetic traces, given some relevant input parameters like the number of nodes, the contacts duration, the inter-contact time, and the node degree distributions (calculated in the previous step). The traces generation process maps each host to a node of the graph and link a pair of nodes with an edge if the two hosts have a potential of becoming in contact (*Potential Contacts Graphs*). The connectivity graph is used as a basis for a *time-varying graph of instant connectivity* for each instant t. In these time-varying graphs (one for each time instant), each link is either *active* if the two hosts are co-located, or it is not present if the two are not. Finally, the generated synthetic traces could be used as test cases for some opportunistic protocols.

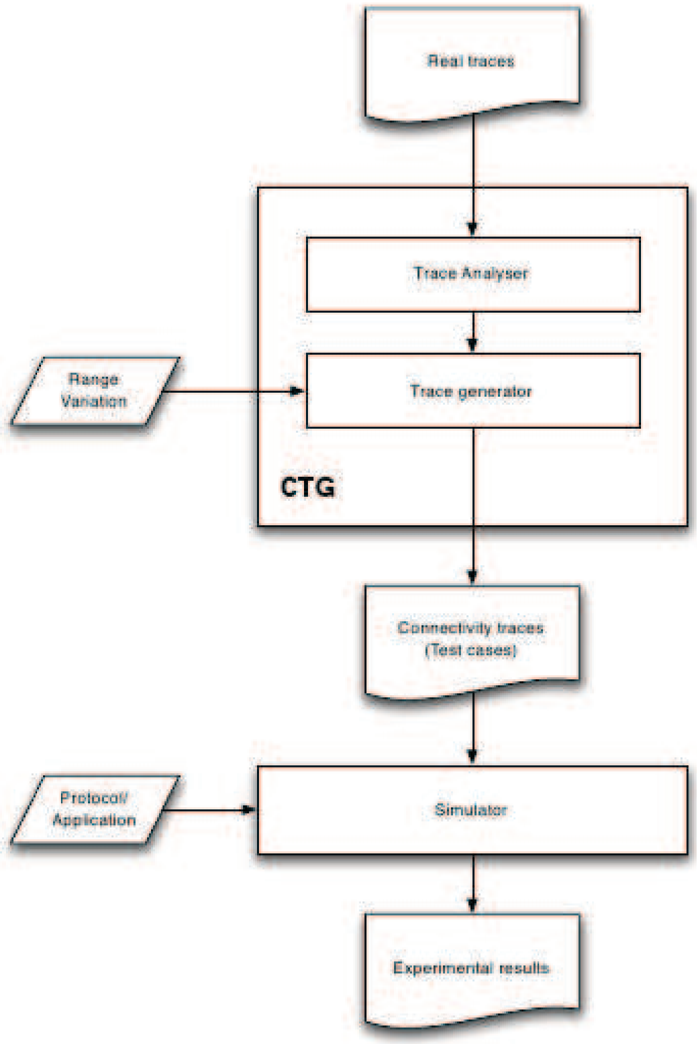


Figure 2.50: Connectivity Trace Generator schema (taken from [25]).

The connectivity model used [24] is build upon two strong (and in some sense not real) assumptions: user's behaviors are *independent* and *uniform* (i.e. one user does not depend of others, and all users have the same behavior). The authors define two random variables X and Y for the duration of the sessions of two generic users a and b, respectively. The probability that a user a will remain in a given location for a time t (i.e., the residence time) is given by a probability density function $p_X(t)$, interpreted as the probability that the residence time will last t seconds. In addition, a probability density function $p_R(t)$ represents the probability that the temporal distance between the beginning of two sessions of two co-located users is t. The objective is to compute a probability density function $p_C(t)$, representing the probability that the co-location (i.e., contact) between any two users a and b lasts t.

Maybe the most important contribution of *CTG*, is the possibility to "scale" the node degree distribution. This feature overcomes the problem that a single set of traces could generate only a single scenario with the same properties, making it possible to test different scenarios scaling the node degree distribution according to: the maximum number of contacts scales up or down proportionally to the square root of the ratio of the total number of vertexes in the graph. The appendix section in [25] describe the methodology of scaling, and an intuitively idea is shown in Figure 2.51 on page 56.

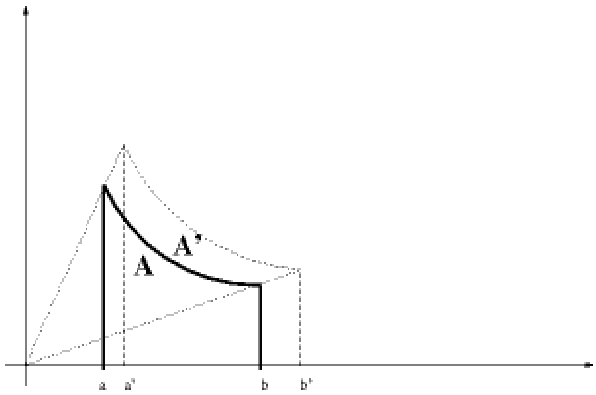


Figure 2.51: Scaling up a graph (taken from [25]).

2.4.8.9 Universal mobility modeling framework

Universal Mobility Modeling Framework (UMFF), proposed in [87], is an effort to improve the *trace-to-model* approach. Authors note that existing mobility models are associated with one specific application scenario, and most of the cases new mobility models are created for every new scenario. In particular, this work has similarities with [75], trying to create mobility models as a group of building blocks.

UMMF enables the *universal* generation of mobility models, based on a small set of fundamental factors, by which mobility models could be classified, and they are: (1) *Target*; (2) *Obstacles*; (3) *Dynamic Events*; (4) *Navigation*; (5) *Steering behaviors* and (6) *Dynamic Behaviors.* A hierarchical diagram of the elements comprising a *UMMF-based* mobility model is shown in Figure 2.52 on page 57 including: (a) a model *environment*: modeled *geographical plane, targets, target sets, obstacles,* and *dynamic events*; (b) a *navigation graph*; (c) a set of *steering behaviors*, to capture the notion of physical forces underlying observed mobility patterns; and (d) scripted *dynamic behaviors*, to influence the execution of mobility models. In addition, UMMF-based models classify agents in classes with specific properties, and group specification.

The most novel element in the hierarchy is the definition of *steering behaviors*, each of who represents a similar physical behavior, where objects moves as result of a combination of forces. These forces where categorized by authors into: 1) *Individual Behaviors* (such as *Seek*, *Arrive*, and *Flee*, which cause agents to react individually to environmental factors or their target selection process); and 2) *Group Behaviors* (such as *Pursuit*, *Interpose*, *Evade*, *Obstacle avoidance*, and others, causing agents to behave in accordance to their relation to other agents, and to the relation

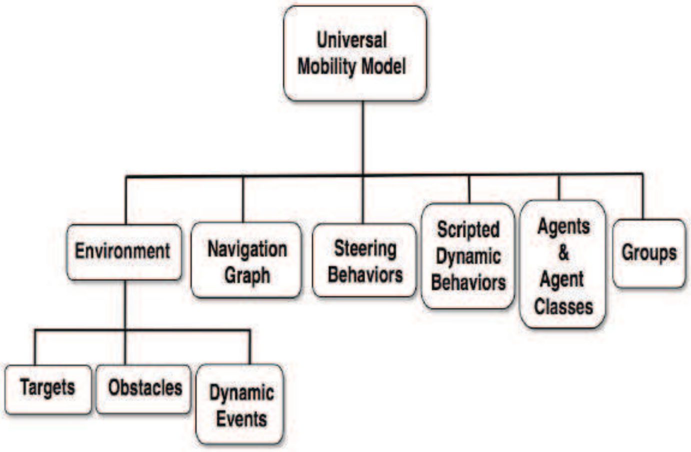


Figure 2.52: UMMF hierarchical diagram (taken from [87]).

between them and the environment). Thus, by combining building blocks is possible to generate complex movements (and then *correlated mobility models*), generating more sophisticated patterns that react to forces from other agents, obstacles, events and its scripted dynamic behaviors. In 2.53 some examples of patterns are shown.

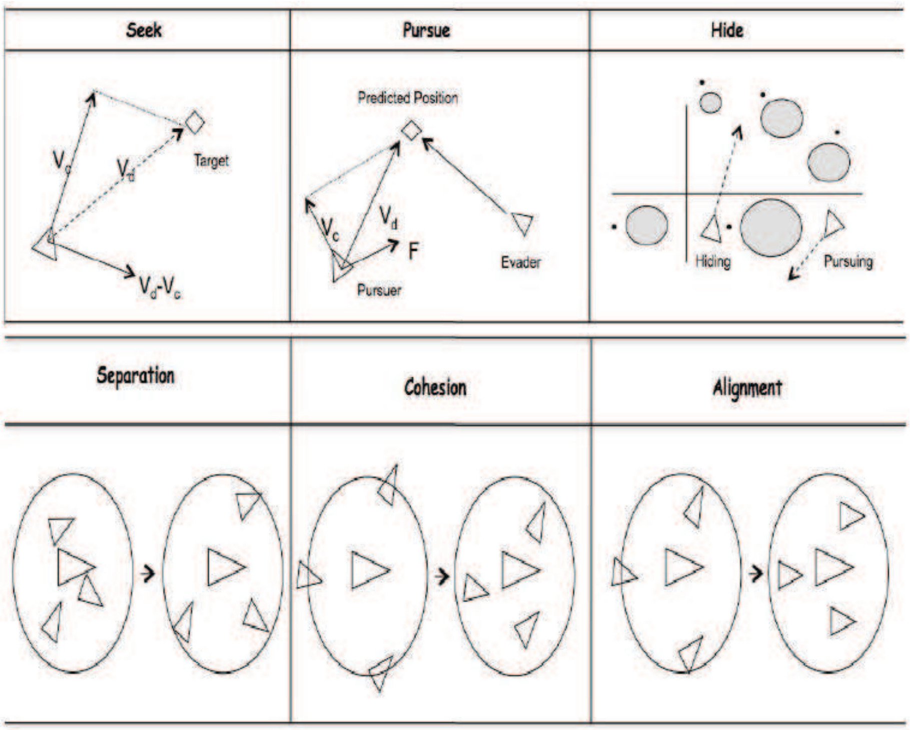


Figure 2.53: Examples of steering forces individually and groups(taken from [87]).

2.5 Mobility models summary

After reviewing in detail each of the most relevant mobility models in the literature in Section 2.4, it is possible to built a table that summarizes the most relevant characteristics of each model. This summary in table form allows the reader to see quickly and easily whether a certain concept of reality is modeled or not, also allowing to carry out a comparison between models. The summary table presents in its first two columns (*Mobility Model* and *Ref.*) the name and reference to the model being described in a given row of the table. The remaining columns refer to the characteristics that are modeled by that model, and the way to indicate that one of them is present in the model is using a √ symbol placed in the corresponding cell of the table. Prior to presenting the table, let us recall what the properties present in the table mean, on which models can be categorized:

- *Random*: it has no dependency of any kind.
- *Temporal*: the next position depends on where the node are.
- *Spatial*: the movement depends on the others nodes near by.
- *Geographic*: the nodes has restrictions imposed inside the simulation area, and could not move everywhere.
- *Hybrid*: the model is compound by two or more of the previous models, to build a more complex model.
- *From Traces*: the model is created from real data, collected from a real deploy.
- *DTN*: if this model was created for *DTNs*, or if it could be adapted to.
- *Obstacles*: if obstacles are modeled.
- *Propagation*: if signal propagation is modeled.
- *On-Off*: if the model models the fact that nodes (or devices) could be in off state.
- *Periodicity*: if the model reflects the human pattern of going back to some places after a period of time.

Mobility Model	*Ref.*	*Random*	*Temporal*	*Spatial*	*Geographic*	*Hybrid*	*From Traces*	*DTN*	*Obstacles*	*Propagation*	*On-Off*	*Periodicity*
Random Walk	*[26, 47]*	√										
Random Waypoint	*[22]*	√										
Random Direction	*[119]*	√										
Random Borderpoint	*[14]*	√										
Clustered	*[78]*	√										
Probabilistic Random	*[31]*		√									
Boundless Simulation	*[26, 50]*		√									
Gauss-Markov	*[77]*		√									
Smooth Random	*[11]*		√									
Random Trip	*[73, 74]*		√									

Mobility Model	*Ref.*	*Random*	*Temporal*	*Spatial*	*Geographic*	*Hybrid*	*From Traces*	*DTN*	*Obstacles*	*Propagation*	*On-Off*	*Periodicity*
Exponential Correlated	*[82]*			√								
Mobility Vector Model	*[6, 69]*			√								
Reference Point Group	*[53]*			√								
Column	*[126]*			√								
Nomadic	*[26, 90, 126]*			√								
Pursue	*[26, 90, 126]*			√								
Ref. Velocity Group	*[117, 118]*			√								
Structured Group	*[17]*			√								
Heterogeneous Group	*[138]*			√								
Freeway	*[6]*				√							
Manhattan Grid	*[6]*				√							
City, Zone, Street	*[85]*				√							
City Section	*[34]*				√							
Weighted Waypoint	*[56]*				√							
Restricted RWP	*[18]*				√							
OMM	*[60, 61]*				√				√	√		
Graph-Based MM	*[127]*				√							√
Area Graph-based MM	*[16]*				√							√
WDM	*[37]*		√	√	√	√		√				√
Disaster Area	*[4]*					√			√	√	√	
Heterogeneous Military	*[80]*		√	√	√	√	√					
UPF	*[83]*		√	√	√	√	√					
CPE	*[67]*		√	√	√	√	√					
Model T	*[59]*		√	√	√	√	√		√	√		
Model T++	*[76]*		√	√	√	√	√		√	√		
Evolutionary Topology	*[86]*		√	√	√	√	√		√			√
Campus Waypoint	*[86]*		√	√	√	√	√		√			√
SUMATRA	*[108]*		√	√	√	√	√		√			√
PCFG Based	*[42]*		√	√	√	√	√					
Connectivity Trace	*[132]*		√	√	√	√	√	√	√			√
GeSoMo	*[38]*		√	√	√	√	√	√	√			√
TVC	*[58]*		√	√	√	√	√	√			√	√
MobiREAL	*[67, 84]*		√	√	√	√	√		√			√
SMM Generator	*[137]*		√	√	√	√	√	√	√			√
MOMOSE	*[21]*		√	√	√	√	√		√			√
CosMos	*[49]*		√	√	√	√	√	√	√			√
MoNoTrac	*[48]*		√	√	√	√	√	√	√			√
IMPORTANT	*[6]*		√	√	√	√	√	√	√			√
CTG	*[25]*		√	√	√	√	√	√	√		√	√
UMMF	*[87]*					√	√		√			√

2.6 Mobility metrics

Mobility metrics are used to measure, characterize, and compare mobility models or to compare models with real data. this metrics could be classified in groups (as proposed in [6]): *(1) protocol independent* and *(2) protocol performance metrics*, and *protocol independent* metrics can be further distinguished into sub-classes: *(a) pure movement* metrics and (b) *link based* metrics and (c) *connectivity graph. Protocol independent* metrics (or *pure movement* metrics) are calculated using node mobility and reachability between nodes, while *link based* and *connectivity graph* metrics are calculated over links. A link between two nodes a and b exists if node a is within the communication range of b. On the other hand, *protocol performance metrics* measures the impact on the performance of ad-hoc networking protocols (e.g performance analysis of *end-to-end throughput*, *control overhead*, *data packet delivery ratio*, *end-to-end delay* and *average hop count*). In [41] a good summary of mobility metrics is presented, including the most used ones defined in [6, 41, 62, 81]. A brief description of some metrics is presented next.

Degree of spatial dependence: measure the similarity of velocities and directions of two nodes (i and j, that are inside of the transmission range) in a certain time (t).
$D_{spatial}(i,j,t) = RD(\vec{v_i}(t), \vec{v_j}(t)) * SR(\vec{v_i}(t), \vec{v_j}(t))$ where $\vec{v_i}(t)$ and $\vec{v_j}(t)$ are the velocity vectors of nodes i and j at time t. *Relative direction (RD)* and the *speed ratio (SR)* between the two velocity vectors are defined as $RD(\vec{v_i}(t), \vec{v_j}(t)) = \frac{\vec{v_i}(t).\vec{v_j}(t)}{|\vec{v_i}(t)|*|\vec{v_j}(t)|}$ and
$SR(\vec{v_i}(t), \vec{v_j}(t)) = \frac{\min|\vec{v_i}(t)|,|\vec{v_j}(t)|}{\max|\vec{v_i}(t)|,|\vec{v_j}(t)|}$.

Degree of temporal dependence: measure the similarity of the velocities of a node at two time slots (not too far apart). A high value means that the node travels in more or less same direction and almost the same speed over a certain time interval. A similar metric is the averaged value over nodes and time instants $D_{temporal}(i,t,t') = RD(\vec{v_i}(t), \vec{v_i}(t')) * SR(\vec{v_i}(t), \vec{v_i}(t'))$

Relative speed: measures the difference of the speed of two nodes i and j in a certain time t
$RS(i,j,t) = |\vec{v_i}(t)| - |\vec{v_j}(t)|$ limited by $Di,j(t) > c * R \Rightarrow RS(i,j,t) = 0$ where c is a constant, R is the transmission range and $RS(i,j,t)$ is the average relative speed, over node pairs and time instants satisfying $\overline{RS} = \frac{\sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{t=1}^{T} RS(i,j,t)}{P}$ where P is the number of triple (i,j,t) that $RS(i,j,t) \neq 0$.

Geographic restrictions metrics: is expressed through the *degree of freedom* of a point, which is the number of directions a node can move after reaching that point and is usually connected to a map. In this sense, this metric is more suitable for models like *Manhattan* and *Freeway.*

In DTN and in ad hoc networks in general, it is useful to have metrics which can analyze the effect of mobility on the *connectivity graph*, because they relate the mobility metrics with protocol performance. The *connectivity graph* is the graph $G = (V, E)$ such that $|V| = N$ and at time t, a link $(i,j) \in E$ if $D_{i,j}(t) \leq R$, where N is the number of nodes and $D_{i,j}(t)$ is the Euclidean distance between nodes i and j at time t. The *indicator random variable* $X(i,j,t)$ is defined as $X(i,j,t)\begin{cases} 1 & D_{i,j}(t) \leq R \\ 0 & otherwise \end{cases}$ so, $X(i,j,t) = max_{t=1}^{T} X(i,j,t) = 1$ indicates that a link exists between nodes i and j. This is the base definition for several connectivity graph metrics, such as *number of link changes* and *average number of link changes, link duration* and *average link duration* and *path availability* and *average path availability.*

Number of link changes: for a pair of nodes i and j, it reflect the number of transitions for this pair if nodes when both get connected/disconnected:

$$LC(i,j) = \sum_{t=1}^{T} C(i,j,t)$$

where $C(i,j,t)$ is an *indicator random variable* reflecting that the link between i and j is *"off"* at time $t-1$ and becomes *"on"* at time t:

$$C(i,j,t)\begin{cases} 1 & X(i,j,t-1) = 0\, and\, X(i,j,t) = 1 \\ 0 & otherwise \end{cases}$$

The *average number of link changes* is calculated with averaging over node pairs satisfying certain conditions, i.e.,:

$$\overline{LC} = \frac{\sum_{i=1}^{N}\sum_{j=i+1}^{N} LC(i,j)}{P}$$

Node degree (ND): is a node density measure, and it counts the number of neighbor nodes (if they are in transmission range R) averaged over the total number of nodes *(N)* and in every time instant:

$$ND = \frac{\sum_{t=1}^{T}\sum_{i=1}^{N} N(i,t)}{T.N}$$

link duration expresses the average duration of the link existing between nodes i and j, and its correspondent *averaged link duration.*

path availability expresses the fraction of time during which a path is available between two nodes i and j.

partitions measures the graph connectivity: a value of 1 means that the network is connected at all times, while values >1 indicate the existence of partitions in the graph.

average time to link break measures how often link breaks, which would trigger message exchanges and recalculations at the routing layer.

topology change rate (TCR) is another way to measure the topology dynamics, which is the number of link changes per time unit as observed by a single node.

inter-contact time describes the time elapsed for two nodes i and j from the time t in which they get disconnected (out of range of transmission) and the time t' when they get connected again.

contact duration is the time for which two nodes i and j remains connected. This and the previous metric are very relevant for opportunistic protocols, because inter-contact times reflects when the two nodes will have the opportunity to exchange information, and contacts durations is a measure of how much information would be passed by between nodes.

2.7 Summary

Mobility modeling is a key-developing aspect for network simulations, and it has a great impact on the performance and accuracy of results. It is clear that most of the synthetic models lack real behavior and that is why the *trace-to-model* approach is preferred for some scenarios, even when the task of getting real traces is more difficult.

This thesis reviews exhaustively the state of the art in mobility modeling, describing many models individually and conforming a very complete description of available models, being a good starting point (as few others works [5, 10, 26, 94, 116, 121]) for the mobility modeling study.

There is not a tendency towards generic mobility models. Instead, many particular mobility models and trace generation tools exist. As summarized in one of the first works of a "general meta-model": *"Despite such efforts, our community has not yet reached a state in which accuracy and representativity of defined mobility models can be assumed soundly and ubiquitously"* [87]. As a consequence of the absence of a general model, because in most cases the models are bounded to a specific scenario, new models are being created for almost every new application domain, and turning difficult to review the huge number of models in the literature.

Chapter 3

Model selection

This chapter makes a detailed description of the problem under study and identifies desired characteristics to be modeled in the scenario of a *Plan Ceibal* network used as an already deployed opportunistic network for *DEMOS*. By contrasting the properties identified in reality and those which are offered by each of the models seen in 2, a comparison is made to take the best decision of what model to choose or will provide strong arguments about the need to create a new mobility model for this particular case.

3.1 DEMOS movement characteristics

As mentioned in Section 1.1, the *DEMOS* project attempts to deploy a service collection of environmental data gathered from different kind of sensors and using opportunistic *Plan Ceibal* networks as its transport network to carry information such as air-quality sensors, that are deployed at the living premises of children in environmentally vulnerable neighborhoods as well as at their schools, parks, etc. The environmental data collected by those sensors are transmitted, using opportunistic networking techniques, to the children's laptops as they pass-by during their daily life. Later, using the same techniques, the data is transmitted into a data-collection point at the school and from there to an environment monitoring station (see Figure 3.1 on page 63).

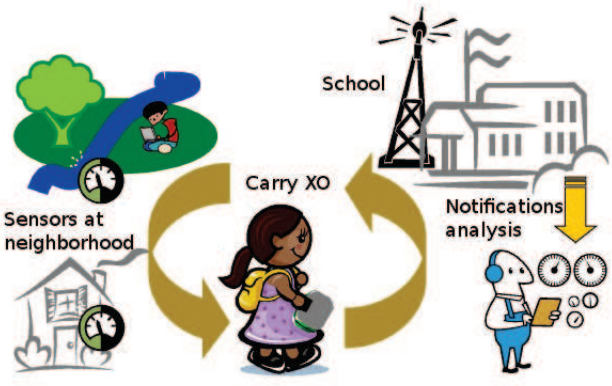


Figure 3.1: The DEMOS Project

The choice of using *Plan Ceibal* networks originates in the fact that they have undergone tremendous growth in equipment and infrastructure deployed throughout the country. Over recent years, *Plan Ceibal* has deployed approximately 570,000 XO laptops and has equipped schools with wireless Internet connection for all scholars inside the school building, turning this scenario into a dense delay tolerant network, in which every XO laptop is a potential mobile node (*MN*) of it. As a consequence of this, a *MN* on a *Plan Ceibal* network is mainly a child with its *XO* laptop, who behave in different ways along the day and depending on the place he is or the time of day, transferring often the same behavior to the XO, considering that it is loaded and moved together with the child. At next, the general movement characteristics of a *Plan Ceibal network* are described, as depicted in Figure 3.2 on page 64.

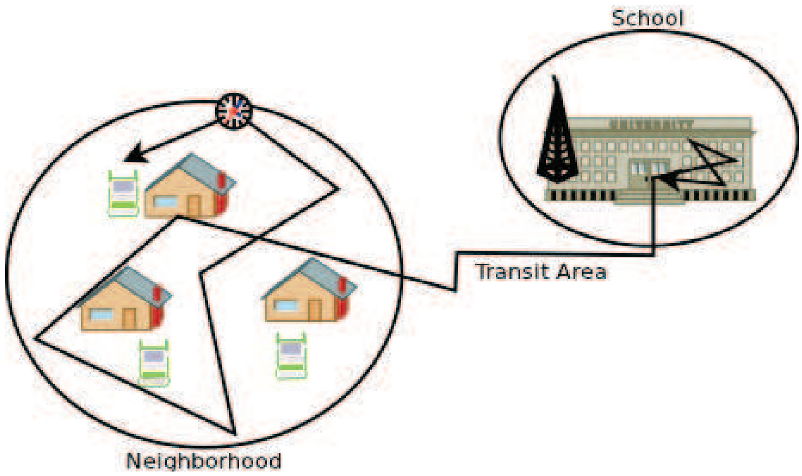


Figure 3.2: Descriptive view of the case of study.

- Children are in their neighborhood and/or home using their *XO* while they move, play with others classmates, stay at home or remains switched-off at home.
- Children transit on the transit area, while they follow its path up to school.
- During class time, inside the classroom and inside the school, classmates moves in a restricted area (inside the school walls). Movements inside the school includes staying in its classroom, and random or in groups movements, always inside the building.
- When classes end, children return to their houses. This movement path is also recurrent, because scholars repeats it every day.
- In many cases, *XO laptops* remain turned off, for example during the night when child sleep, and this is a relevant behavior to take in consideration, because of (1) in *DTNs* it has a big impact on the opportunity to exchange information, and (2) because notebooks, as opposite to *PDA* or mobile phones, are not always-on and people does not use it frequently while they are moving.
- Since *DEMOS* is concerned to study environmental information, in addition to laptops, the real scenario should have sensor devices spread out of the area under study that do not belongs to the *Plan Ceibal* network itself and having a fixed position all the time (they will not be mobile nodes).

After an analysis of the characteristics of movement patterns that must be modeled, the key aspects to be considered are:

- Movements have spatial and time variants. People move in different ways depending on where they are and the time of day, showing *temporal dependencies* and *geographic restrictions*.
- Nodes, in general, does not move on all area, but they have some predefined sub-areas instead, that restrict them in their movement (*geographic restrictions*).
- Real life presents both individual and group behavior, even for the same person. This leads to movements with *spatial dependencies* and combination of way to move (*hybrid* characteristics).
- Nodes have *on-off* periods.
- In general, people behavior present a daily based periodicity.
- Even when *DEMOS* is concerned to the study of sensor networks to transport information up to schools and back, sensors will have fixed positions all the time, so they will not be taken in consideration for the mobility model.

3.2 Data availability

An important restriction for this work is about what data could be generated by and collected from *XO* laptops, that are cheap devices with basic hardware with just one wireless network interface. The absence of more complex ways of communication, like mobile phones, *GSM* modems and *GPS* imposes strong limitations to get global positions, making unfeasible the use of certain models. Even when *GPS* devices could be connected to laptop using a *USB* cable, this solution is expensive and is out of scope of the project that looks for the use of the *Plan Ceibal* network *"as-it-is"*. In summary, the activity and connectivity data available to be gathered from laptops are:

- *Sessions*: laptops can report times when it is switched on and turned off, and session duration could be derived from them.
- *Contact duration*: laptops can report times when it starts and stops to interact with another laptop (both are in or are no longer in the range of visibility of each other), and contact duration could be derived from them.
- *Visible Access Points*: laptops can report times when it starts and stops to be in the range of visibility of an access point.

Given that the only known location is the school building and without having option to get accurate data about location from laptops, selection of the model will be strongly influenced by this fact.

3.3 Previous work with DEMOS

DEMOS has produced related work about the problem of content-based routing and its system applications, which have in turn feeding back knowledge and design choices [7]. On mobility models in particular, there are also previous results [44, 115] that have helped the project to make some decisions about which tools to use in order to obtain good results at simulation test scenarios, as shown at next:

- Early work in *DEMOS* with network simulators were made to use *NCTUns* [122], which at that time had several strengths that placed all together made it the perfect simulation tool for the project when compared to other network simulators. One of the requirements to become the simulator to be used is to be free and open source software, and *NCTUns* fulfilled that requirement (like others). Due to the fact that the software developed for *DEMOS* should run in certain environments and platforms, *NCTUns* also allowed important things that made the difference for its choice, such as allowing simulation and emulation of software and also enabling software that has been developed for *GNU/Linux* to run without any change in the program's source code inside the simulator. Besides the above advantages, *NCTUns* provides direct emulation over *GNU/Linux*, with extra functionality for the project as the use of an existing 802.11s [52] standard implementation for *GNU/Linux* like [109] (the same protocol than the one used by *XO* laptops). As a result of this stage, the *Demos Mobility Model* [44] (*DMM*) was developed, built as a new mobility model for the *BonnMotion [3]* mobility trace generation tool, and an application to export generated traces by *DMM* to the *NCTUns* format was developed.

- Later, some problems identified with NCTUns and other events led to review the decision of which network simulator to use: the 802.11s implementation of GNU/Linux did not work as expected, NCTUns began charging licenses for use, Plan Ceibal stopped using 802.11s in XO laptops, and the research group made a contribution [99] to the *ns3* [101] network simulator, which can run software that runs on *GNU/Linux* without modification. From then on ns3 became the official network simulator for the *DEMOS* project.

- Finally, a new addition [129] *to ns3* simulator was made that allows loading movement traces in *ns2* [100] format inside *ns3*. The main objective of this contribution was to use the *DMM* model in *ns3*, but it also allows: (i) *ns2* simulator users (widely used in research by academia) can use *ns3*, (ii) mobility models which are not yet implemented in *ns3* can be used, and (iii) tools that generate traces of movement only for *ns2* (as *BonnMotion* [3], *SUMO* [9] and *TraNS* [111]) could be used.

3.4 Selection of the model

At the time when (1) a detailed study of main characteristics and restrictions of reality was made (in Sections 3.1 and 3.2), (2) after a summary of the previous work with DEMOS was presented and (3) after a good comparison of the mobility models was presented (in Section 2.5[1]), we had all the relevant information to decide the model that better adapts to the case of study.

- Although *DMM* [44] attempts to reflect important features of reality such as daily periodicity and it can generate traces for different scenarios by changing parameters, it has not been validated against real data. Even when it was built based on the basis of intuition, the model does not reflect an important feature for a school environment such as weekend days, for which it is reasonable to assume that the mobility pattern is different from weekdays. Moreover, the synthetic model defined in [115] is able to generate scenarios that can handle certain metrics established beforehand, but does not model the daily periodicity nor the weekend aspects of the real scenario. Thereby, it may be taken as a very useful model for testing the opportunistic protocol in many situations, but away from being the suitable model to the particular scenario under study. Last but not least, none of the two models reflects

[1]For this work, *obstacles* and *propagation* properties of the models are not relevant, because they could be managed by network simulators, but they were left intentionally for further reference in the table.

the *On-Off* behavior, identified as one of the main properties that affect the scenario under study.

- The use of the classic *synthetic* models for modeling human real scenarios has been widely discussed in the literature, because of their simplicity and lack of many real characteristics of human behavior.
- Regarding the *hybrid* models, even when they intend to represent more complex scenarios, none of them reflects all desirables characteristics, or directly, they where built for specific scenarios like military [80], disaster areas [4], pedestrians [83], among others.
- Most of the *trace based* models require the usage of detailed trace paths of nodes to get from them the patterns of movements. As said in Section 3.2, this resource is not available in the Plan Ceibal network scenario.
- The *WDM* model [37] seems very close to our needs, because it implements very well the human periodicity (people going an back to work or in our case, to school) and reflects the use of vehicles. The main disadvantages in the use of this model for this thesis are derived from its microscopic approach. Accurate results implies to have detailed location data (like routes, number and frequencies of buses and cars, home addresses and points of interest), information that is not available. The model also has the drawback that it does not implement the *On-Off* property.
- Looking models that specifically implements the *On-Off* property:
  - *Disaster Area* [4], as mentioned above, was created for a very specific scenario.
  - *TVM* [57] has the appropriate characteristics, but the calibration needs some parameters that depends on data not available in this work, like speed of nodes.
  - *CTG* [25] uses connectivity information (like contacts durations between nodes and inter contact times) to generate mobility traces. This approach seems to be the perfect option for this work, by using information about connectivity instead of microscopic level of details like positions or speed.

3.5 Summary

For the problem under study, and bearing in mind the real data available from laptops, the use of a *Connectivity Model* approach is more appropriate than the use of a pure *Mobility Model* without using detailed data about locations of nodes, maps of the cities, routes of public transport or any other information about mobility at microscopic level. *CTG* provides a mechanism to study and model a real scenario using statistical data obtained from the real deployed network for the configuration of the trace generator. As a big difference, the *On-Off* property is implicitly included in *CTG* while in the new model it will be explicitly modeled, generating *On* and *Off* events for nodes.

Chapter 4

Real network data

4.1 Objectives

After selection of a suitable model, the next step was to configure it properly, tuning carefully its set of parameters. Even when there exists many real traces from a wide range of scenarios [135], data for this particular scenario is not publicly available yet. Given that a real deploy of *Plan Ceibal* networks exists, it is possible to collect data from a already deployed network. The objectives of this section are:

- To describe the kind of data to be collected, to list the constraints of the environment that will restrict the design and implementation decisions and to describe the software tool to get the data.
- To show a methodology to derive some connectivity metrics needed by the model.

4.2 Collecting information

User activity was collected using a client-server application developed ad-hoc for this purpose, with the support and supervision of Plan Ceibal [107]. Due to the fact that the network is "in service", this software should have negligible impact on performance. Laptops have very low hard disk capacity and therefore is not possible to maintain logs for all period of collection of data; consequently, the information should be uploaded to a central repository whenever is possible and erased as soon as possible. Finally, the software must run in user mode, without root privileges. All the information is propagated and gathered in the school server, located inside the school building.

Every laptop must be uniquely identified, and data should refer to all day activity, and not only inside school. Since laptops are not always connected to the school server or to the Internet, the software application should be able to maintain the off-line data until it could be uploaded and then deleted from local storage on the *XO* laptop.

The gathered information must permit to empirically estimate statistical distributions for:

- *On-Off* time
- Number of laptops on the range of another laptop (*node degree*)
- How much time two laptops remains connected (*contact time*), and

- Time elapsed between a node disconnects and reconnects another once (*inter-contact times*).

4.2.1 Software and platform

The data collector was built on top of another tool developed by *DEMOS Project* called *RON* [7]. *RON* was designed to be a flexible software to be used in a resource limited devices, following a *DTN* approach, that meets the requirements of design. It is a framework that allows routing generic information along an opportunistic network, implements checks of duplicated, has replacement policies, manages the amount of information maintained for a time period, and is fully configurable. With *RON*, the problem of routing messages from source to destination is solved.

The connectivity information is generated by a set of scripts running in *XO* laptops under unprivileged user as a daemon while laptop is on. These scripts passes data to the *RON* software, who routes the information to the school server, who receives and persists the information in a database. The school server runs the *RON* software too, but using a different configuration and with another set of scripts (different from the scripts that runs in laptops) because it works only as a collector of information and not as a source of information. Specific software was also developed for the school server, as for the *XO* laptops to be connected to the *RON* process and to persist the arrived information in the central database.

4.2.2 Connectivity information

The connectivity information managed by the system, which would be used to derive statistical distributions for certain metrics, is defined below:

KEYID: Every node in the network is identified by its serial number, which is unique across the entire network. Every message sent by the software includes this information to identify the node who is originating the message.

(START/SHUTDOWN)TIME: When a node is started up, it informs about its startup and last shutdown time. This events are reported using a local timestamps of each node.

PING: Because start and shutdown times are relevant for the estimation of On-Off distribution, each node sends periodically this information in a single message, including both times. This behavior intends to minimize the lost of this information, due to occasional loss of messages, which is characteristic of opportunistic networks.

NEIGHBORS: For the calculation of connectivity metrics, each node reports about others laptops around it. Periodically, the software sense the wireless network and checks if the neighbor list at this time differs from the last time it was checked. In case of changes, laptop informs the new list of neighbors and the local time when this change is detected.

APS: Similar as *NEIGHBORS* notification, each node informs about the visible access points. Even when this information is not enough to estimate global positions (because the only access point of reference is the school) it may help to derive some behavior parameters refereed to periodicity and recurrence to the same places.

4.3 Data analysis

The connectivity data collected by the system was post-processed in order to get accurate statistical information, and some information was pruned: (1) the first two weeks of data contains no useful data, because the system was not in its steady state, (2) some nodes had shown inconsistent time data, probably because errors in hardware that makes the events were reported with wrong timestamps, and (3) some nodes had reported more than one host key, probably because user made a system upgrade. Finally, all results in this chapter are calculated using data from 118 *XO* laptops over a period of 50 days.

4.3.1 On-Off

The first characteristic to be studied is the *On-Off* behavior of nodes. Following intuition, we could say that nodes have differentiated pattern along the day, and in particular, that the activity decay during the night, and increases during the day (see Figure 4.1 on page 70). This simple observation lead us to the conclusion that this kind of events should be used in different ways for each hour of the day.

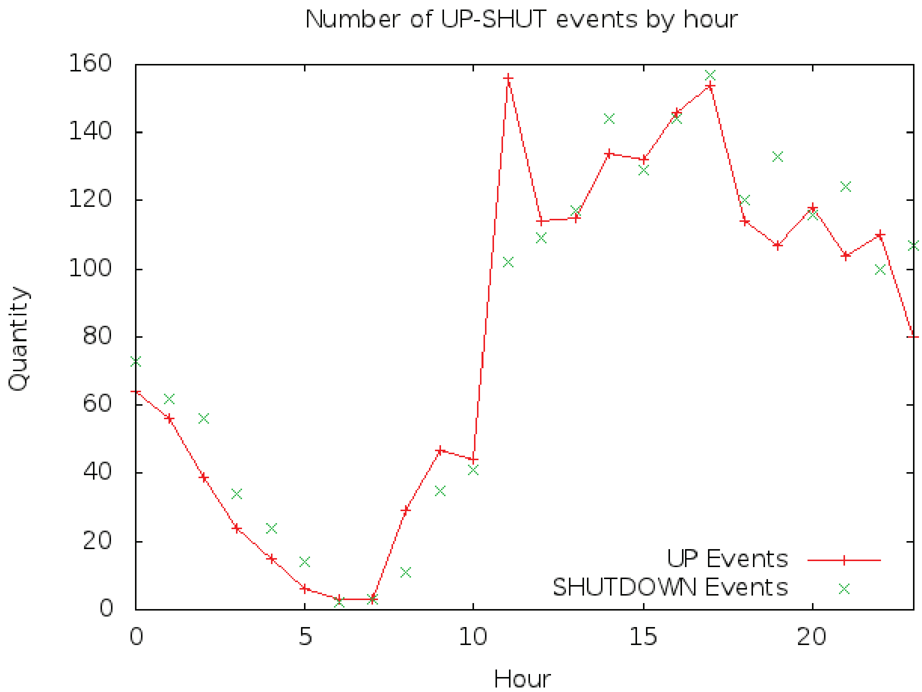


Figure 4.1: Number of events for a set of traces (UP-SHUT) by hour.

Figure 4.2 on page 71 presents the frequency and cumulative distributions of session durations (time while a laptop remains turned-on or turned-off). Both graphics show that is very frequent to have short periods of time when laptops are up or down, but the cut-off times are different. For the uptime durations, 90% of cases, laptops remain on up to 20000 seconds (approximately 5 hours and a half). Several factors influence this result, for example, the lifetime of laptop battery and the use of laptop before, during, and after school (remember that it is possible the user turn off the device before moving). On the other hand, the analysis of the shutdown times shows that, even when there is an important number of turned off periods shorter than 20000 seconds, 90% of the cases have a duration less than 80000 seconds (approximately 23 hours). As seen in Figure 4.2 on page 71 too, it presents a more dispersed distribution than the uptime, meaning that time periods of off times are more probale. This big difference is explained largely by the long periods in the night, when laptops remain off.

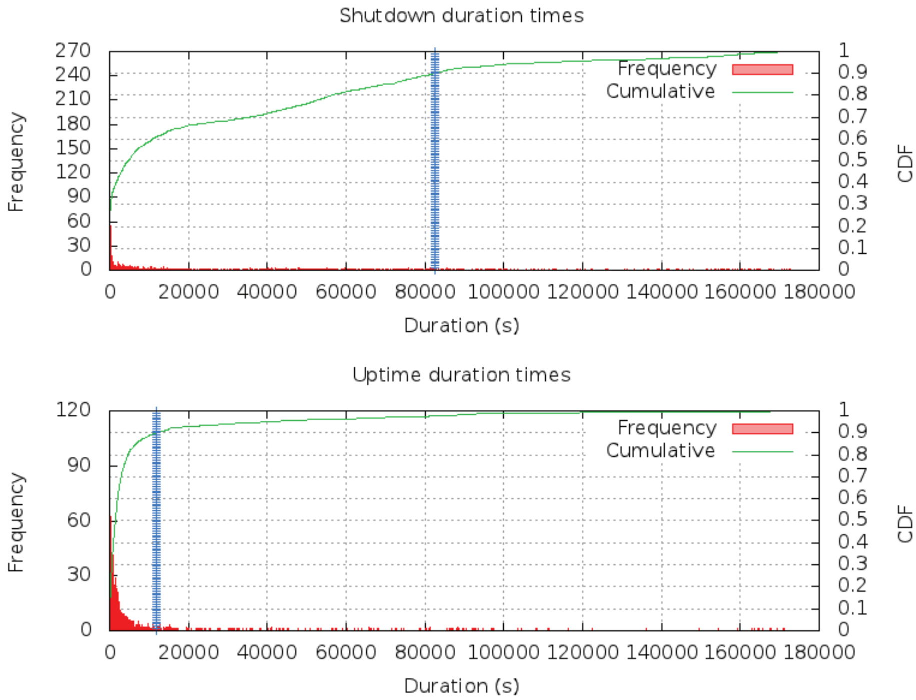


Figure 4.2: Frequency and cumulative distribution of SHUT and UP time durations.

Contrary to what intuition might indicate, it is interesting to note that the distribution of times does not change between working days and weekends (see Figure 4.3 on page 72). This aspect has a great impact on the model, because it permits modeling the same *On-Off* behaviors for any day of the week.

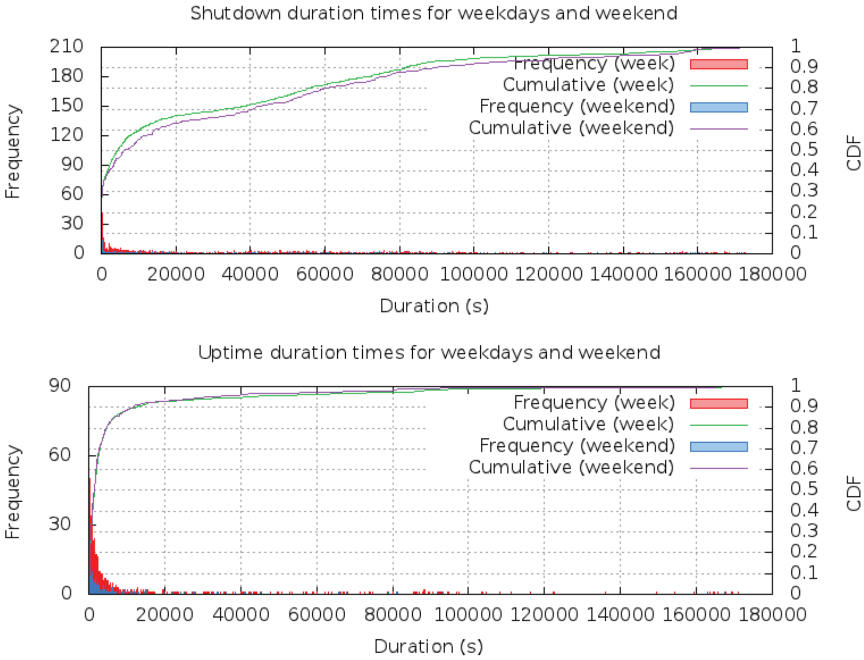


Figure 4.3: Frequency and cumulative distributions by days.

Finally, Figure 4.4 on page 73 and Figure 4.5 on page 73 show the distribution of time duration while a laptop remains on and off. The data was divided into two time periods: (1) from 8 a.m. in the morning up to 18 p.m. in the afternoon, and (2) from 18 p.m. up to 8 a.m. of the next day. The durations of the shutdown times are different in two cases, reflecting that within the school hours is more frequent to have minors times off than those out from the classes schedule. This behavior corresponds to the intuitive idea that children have to use laptop in classroom, but also because outside school hours there are factors such as night hours, with much less activity and thus having long switched off times. On the other hand, uptime durations are not much different depending the time of the day, meaning that in most of the cases and independently of the time of the day, children works on its computer on short periods of time (see that the 90% of the user sessions last for at most 2.7 hours).

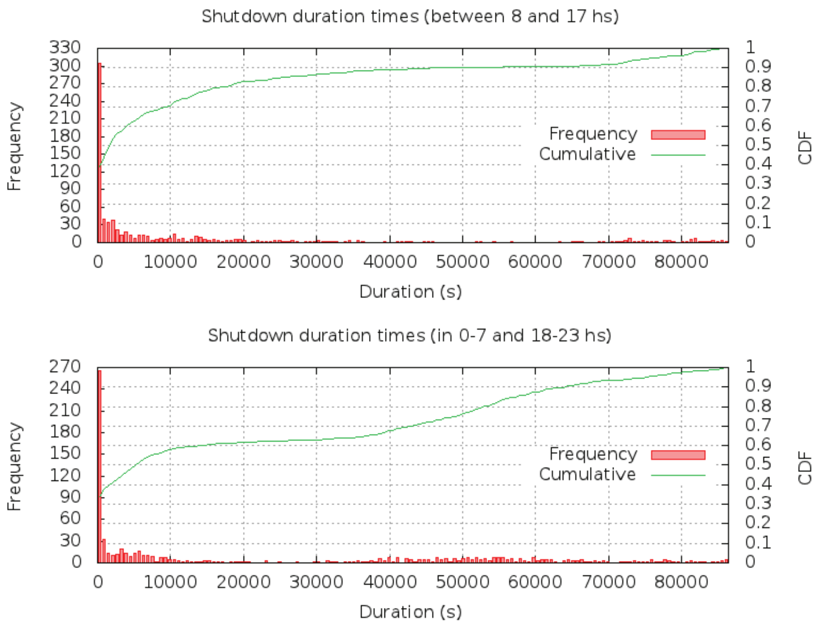


Figure 4.4: Frequency and cumulative distributions for shutdown duration times by hours, within classes hours and outside the school.

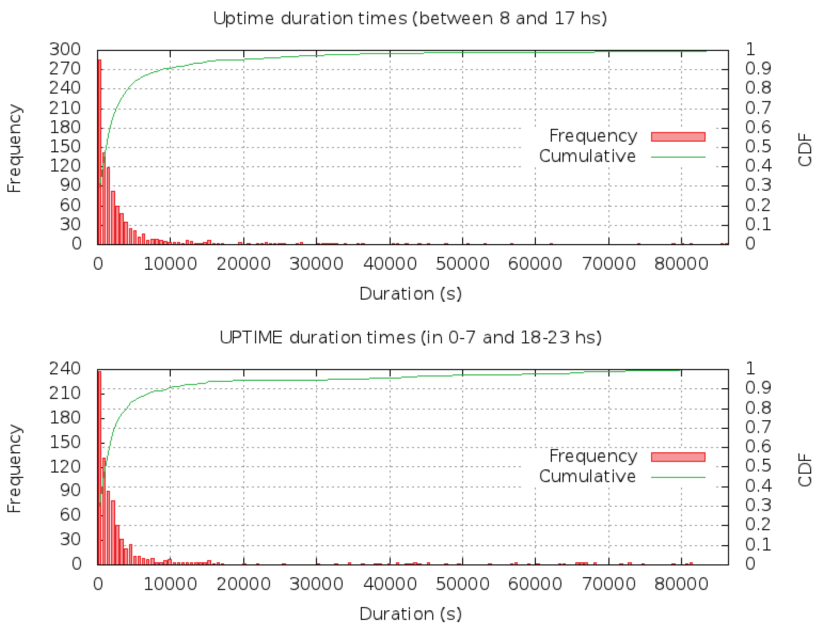


Figure 4.5: Frequency and cumulative distributions for up duration times by hours, within classes hours and outside the school.

4.3.2 Contact durations

Contact duration is measured as the time that a contact between two nodes lasts, as shown in Figure 4.6 on page 74. The distribution of *contact duration* (Figure 4.7 on page 74) shows that most of the cases, the contact of two laptops lasts for less than 2500 seconds (approximately 40 minutes). However, an important fact that was not really intuitive at first, is that there is no contacts between laptops during the weekend, even when they present On-Off activity (as shown in 4.3.1).

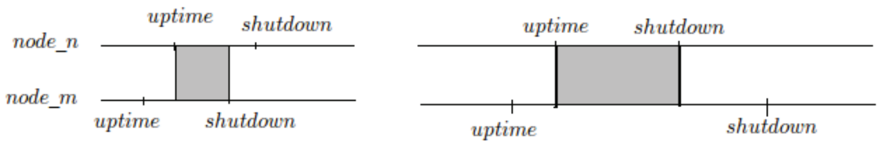


Figure 4.6: Situations where contact duration time exists.

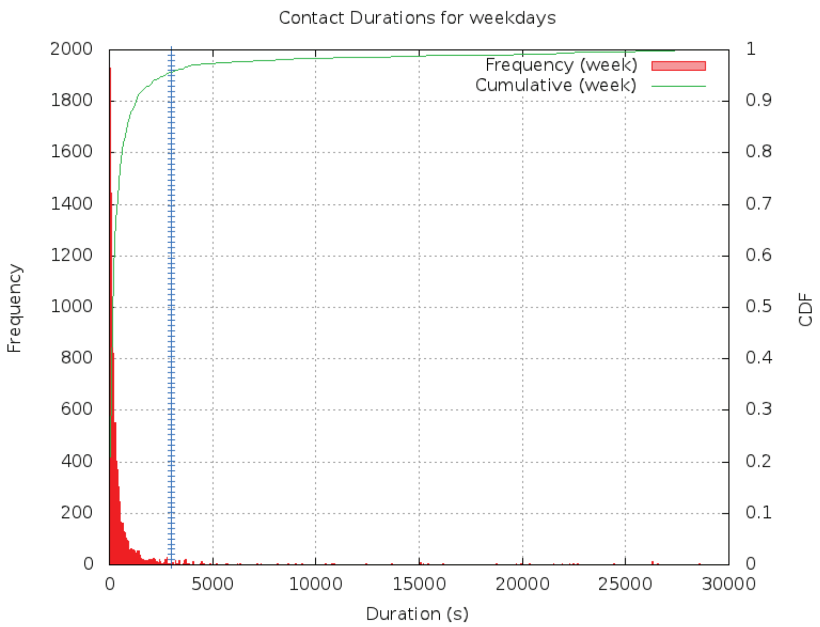


Figure 4.7: Distribution of contact durations.

4.3.3 Inter-contact time

The distribution of *inter-contact* time represents the time elapsed between two laptops from where they ceased to be connected, until they become connected again (see Figure 4.8 on page 75). Figure 4.9 on page 75 shows that in most of the cases, two consecutive contacts take place in the next four hours after the disconnection time. Another characteristic shown is that for those laptops that remain disconnected for more than four hours, they will be connected again in a time range of days, and for this reason the metric shows accumulations around the integer values of the x-axis. Intuitively, this fact means that laptops have a periodical reconnection, which corresponds to the connections generated in school, if not every day, then the next day that they were near one from another and inside the school.

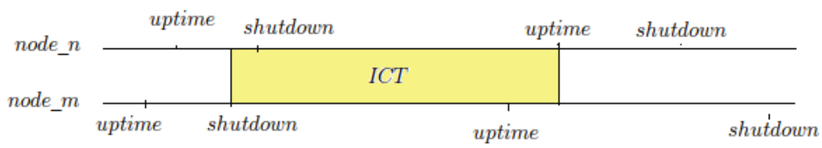


Figure 4.8: Example of a calculation of ICT between nodes.

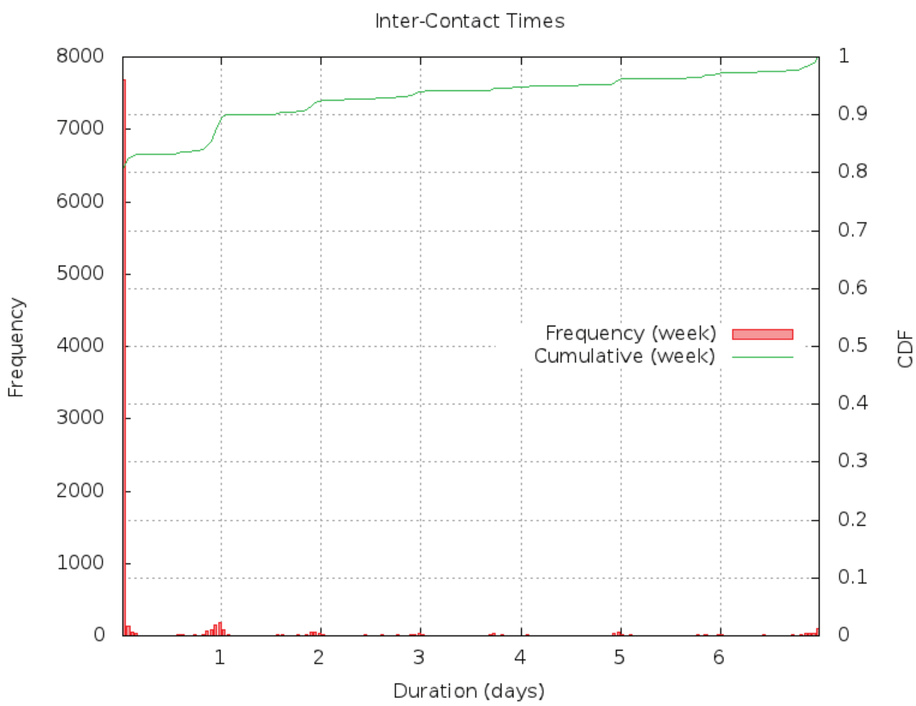


Figure 4.9: Distribution of inter-contact time.

4.3.4 Locations

As noted in Section 3.2, the gathered data only includes information about the list of visible access points from the *XO* laptop (see 4.2.2), and the goal is to estimate how much the user move, or to determine when the user is quiet (or at least that it maintains within the coverage area of the same access points) while the laptop is on, instead of try to exactly know where the user is all the time. Before presenting the results, the model and some definitions needed to understand how data was processed is described:

- A session S represents the time while a laptop remains on (i.e. between an *UPTIME* and a *SHUTDOWNTIME* notification). It may be understood as the time during a children is working on the computer.
- AP_s is the amount of different APs seen during a session from a laptop.
- $PrevAPs$ is the set of APs seen in the present APS notification.
- $PresentAPs$ is the set of APs seen in the present APS notification.
- $newAPs$ is the set of APs seen in the present APS notification that were not seen in the previous APS notification.
- $missingAPs$ is the set of APs seen in the previous APS notification that were not seen in the present APS notification.
- $recurrentAPs$ is the set of APs seen in the present APS notification seen in the past for the session.
- $TotalAP = \sum AP_s$ is the sum of all AP_s seen over all sessions of a node. It might count duplicated access points, and $TotalAP \geq AP_s$ is always true.
- $UniquesAP$ is the amount of different APs seen over all sessions of a node, note that $UniquesAP \leq TotalAP$.
- I_s (*Invariant APs session*) is the set of APs that remain visible during all the user session, note that $I_s \leq AP_s$.
- TI (*Total Invariant*) is the set of APs that remain visible over all user sessions, having that $TI \leq I_s$.
- D_i is a numeric metric (the distance between sets) that measures the number of changes of APs (and in consequence of location), trying to reflect the impact on mobility derived from differences between two consecutive APS notifications. D_i is defined as a weighted sum of partial AP_s up to the moment of the current APS notification, *PresentAPs* and $PrevAPs$, and reflects that not every change (an access point appears or disappears) has the same global impact on the location:

$$D_i = \begin{cases} \mathbf{0}\ when\ PrevAPs \subseteq PresentAPs \vee (PrevAPs \cap PresentAPs) \neq \emptyset & (1.1) \\ \mathbf{0} \quad when\ I_s \neq \emptyset & (1.2) \\ \mathbf{1} * (\#\mathbf{newAPs} + \#\mathbf{missingAPs})\ when & \\ \quad (PrevAPs \cap PresentAPs) = \emptyset\ \wedge PresentAPs \neq \emptyset & (1.3) \\ \mathbf{0.1} \quad when\ PrevAPs = \emptyset\ \vee\ PresentAPs = \emptyset & (1.4) \\ -\mathbf{0.1} * (\#\mathbf{recurrentAPs})\ when\ I_s = \emptyset\ \wedge\ AP_i \subset AP_s & (1.5) \end{cases} \qquad (1)$$

In cases (1.1) and (1.2) there are no significant differences with the location information, because the node still is seeing the same core set of APs. Equation (1.3) defines a high value for D_i, denoting that there are relevant changes in the location information (none of the APs in the previous notification remains present in the next notification). Equation (1.4) is used when a transition exists the node no longer sees any AP or ehen a node that in the previous notification has no APs within the range of visibility. Finally, Equation (1.5) is for cases when a node without any visible AP begins to see an AP already seen before. This last case is intended as a fact that affect in a positive way, because the node is back on the same site. C_s is the cumulative sum of all the D_i values calculated over the session: $C_s = \sum_s D_i$.

Figure 4.10 shows the relation between $UniquesAP$ and $TotalAP$. Both values are the same for a node only when it does not move at all or when a node and all its visible APs were moving all together all the time (in our study we assume the first without lost of generality). On the other hand, $UniquesAP < TotalAP$ means that the laptop was in different places, and that some of the APs were seen more than once. This is a very important point, because it reflect the fact that children uses their laptops only in a few places, and even more, they have a strong recurrence.

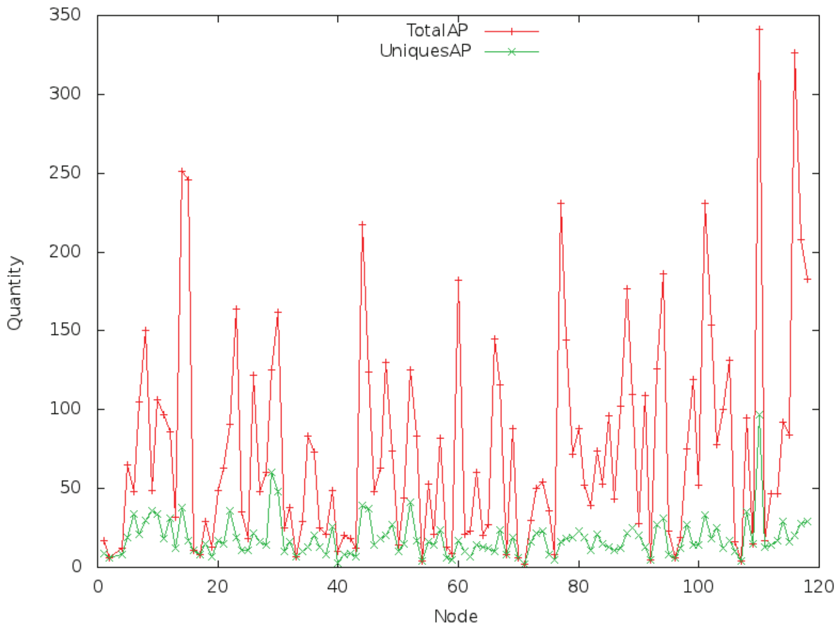


Figure 4.10: Relation between APs.

The data in Figure 4.10 on page 77 shows that 83% of the sessions have $I_s \neq \emptyset$, meaning that the user does not change their position drastically during the session. For the remaining 17% where $I_s = \emptyset$, the use of C_s is needed in order to quantify the location change. Most of these cases had values of $C_s < 1$ and they represent cases when sets $PresentAPs$ and $PrevAPs$ change from an empty set to a set containing just one AP, showing maybe that an AP was restarting or that signal would be weak, but even then it will be considered as the user was at the same place. In summary, 98.8% of the user sessions should be taken as sessions where the user does not move while he is working on his computer.

Another interesting result is that invariants sets for almost all users have $TI = \emptyset$, even when all $I_s \neq \emptyset$ for all user sessions. This result is related to the fact that users have a recurring behavior, going from their houses to the school and back every day. This condition implies that $TI = \emptyset$ because the I_s from user home is different of the I_s in the school, and then the intersection between School APs and Home APs is null.

A study of the name of APs has shown that access points seen outside of the range of the APs schools are different for all nodes. This gives the idea of isolation of classmates when they are not in school, and even when they have contacts with other children, they do not share locations with another students from the same school.

4.4 Summary

This chapter shows a detailed description of the data collected (and the software developed for this purpose) for the movement and connectivity characterization of the real network, and its subsecuent processing and characterization of reality. In this way, it has proven that both mechanisms are viable and permit, in an easy way, to repeat the study on similar scenarios. Statistical information about the metrics are met, and good movement characterization was done.
The main features of user movement discovered are:

- Classmates from the same school have strong interaction when they are inside the school.
- When users are at their homes, there are no interactions with classmates from the same school, but with others classmates instead. This fact is an important simplification in order to model this behavior.
- The users have an *On-Off* pattern according to the intuition:
  - they have laptops on for more time during daytime than during the night
  - time periods when the laptop remains off are longer during the night.
- The users do not change their location drastically while they are working on the computer. In our study it will be assumed as if the user was in a fixed location. This fact has a really important impact on mobility, because movements between school and homes are made when the laptop is off and it is not necessary to model the way they transport from one location to another.

Chapter 5

Demos Connectivity Model

This chapter of the thesis defines the formal definition of the proposed Demos Connectivity Model.

5.1 Objectives

The objective of this chapter is to define a connectivity model that follows the main characteristics depicted in chapter 3 and matches the connectivity information derived in Section 4.2, including the input parameter list accepted. The mobility generator tool for the new model has to be able to create events following some distributions, therefore is necessary to show that the software tool is able to generate data that conforms the real data distributions.

5.2 Connectivity model

The Demos Connectivity Model (DCM) is inspired in [25] and after the analysis made in Section 4.2 it can be said that is a particular case of it. The CTG model is a pure connectivity model, but it was defined for a different scenario, making necessary to derive some parameters that are no needed in the particular scenario tackled in this work. The main differences between DCM and CTG are summarized at next:

- The present work does not need to estimate the *co-location probability* because the network under study consists in only one school building and not in a campus, so the dimension of the area is small and it is covered by one access point.
- Due to the fact of the small dimension of school building, it is reasonable to assume that *contact durations* and *inter-contact times* generated inside the school building are strongly tied to the session durations. This assumption is also a consequence of the small dimension of the school building, that makes possible that laptops could be visible with others.
- CTG does not model *On-Off* events explicitly.
- CTG generates connectivity traces without information of locations.

Having in mind the above valid assumptions derived from the real data, and taking note of the new features to be included, DCM is presented below.

5.2.1 Parameters for the model

The parameters needed by the model are: (1) the number of nodes, (2) the distribution of *On-Off* duration times, and (3) the duration of the simulation.

5.2.2 Locations

The locations of nodes and school are disposed as shown in figure 5.1, where: r is the range of coverage of the XO laptop, R is the range of coverage of the school access point, and N is the number of nodes.

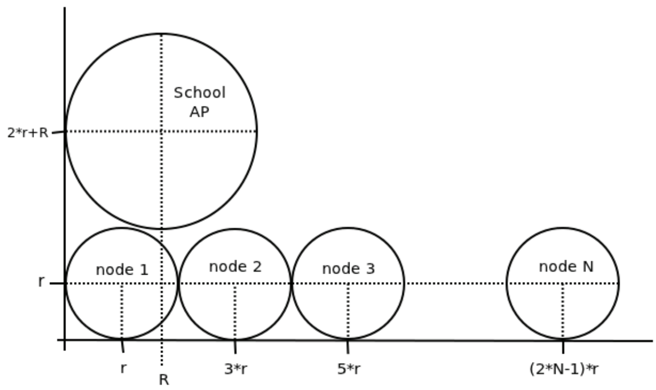


Figure 5.1: Disposition of nodes and school access point.

The location changes for a node alternate in a sequence as follows: $node_i$ starts in position $(x, y) = ((2 * i - 1) * r, r)$ and translates to school at $(x, y) = ((2 * r + R), R)$ and these changes occur when the user arrives to school, and after that, when the user leaves school to go back home. The translations are performed in a time instant instead of modeling the real user path taken to go from one location to another, and this could be done because as derived from traces, the laptop user is off while this transitions take place.

5.2.3 On-Off

On-off behavior is ruled by the same distributions from the real traces that had shown different patterns for distinct hours in a day. The model uses one distribution for uptime duration, since there are no significant differences depending on the hour time, and two for the shutdown duration times, since they are very different (see Section 4.3.1 on page 70).

The sample data of uptime was fitted by a Power Law distribution (see figure 5.2) using [113] and in particular the *plfit* function [33] from the work [32] to estimate the fitness with a Power Law. The *plfit* function is an enhance for Power Law distributions using the *maximum likelihood estimation* method [45].

The distribution of shutdown duration times could not be fitted by any known theoretical probability distribution, and a simple linear interpolation method was used to simulate it, using the real data itself. Figure 5.3 shows the estimated frequency histogram and the cumulative density function for this simulated generator, showing similarity with the real data.

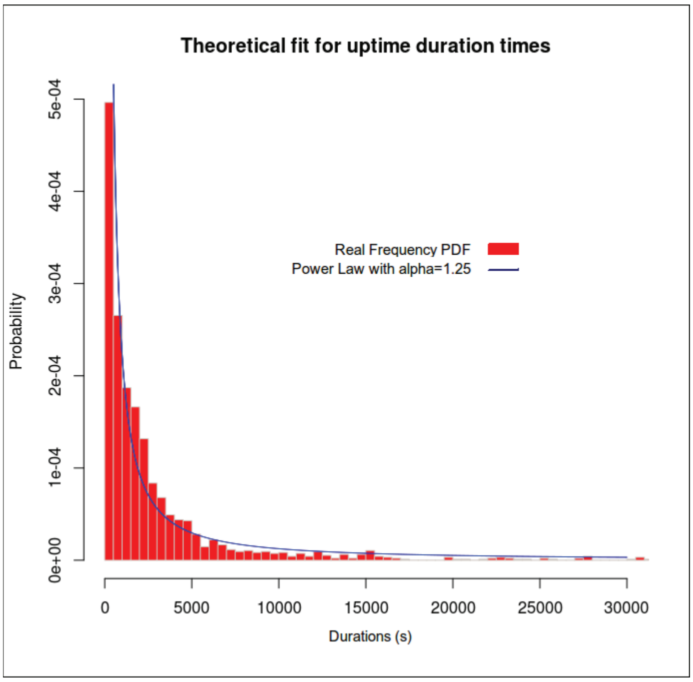


Figure 5.2: Fitting of Uptime durations.

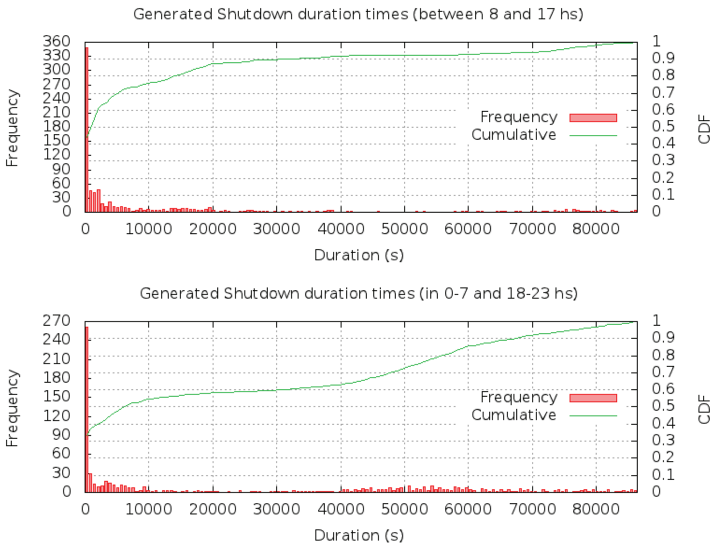


Figure 5.3: Estimated linear interpolation of shutdown duration times.

Figure 5.5 and Figure 5.4 present the comparison of the real data distribution against the empirical distributions of data created by the generator, showing that the real information gathered from school is well fitted.

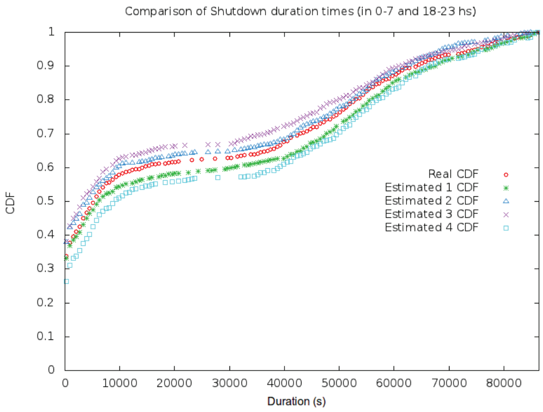


Figure 5.4: Comparing Real distribution vs built distribution generator.

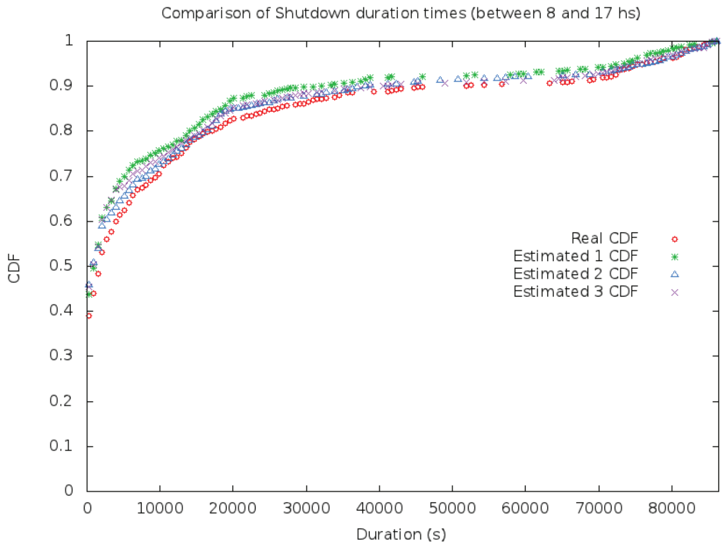


Figure 5.5: Comparing real distribution vs data generated by the built distribution generator.

5.2.4 Contacts durations

The contact between laptops happens mainly when users are in the school and since the school building has a small area dimension, it is reasonable to assume that if two laptops are inside the school, both of them will be in contact. This fact allows to assume that contact durations inside the school have a strong correlation with the *On-Off* pattern, it can be assumed as the same.

Sample data of contact durations was fitted by a Power Law distribution (see figure 5.6) using [113] and in particular the *plfit* function [33] from the work [32] to estimate the fitness with a Power Law and its exponent.

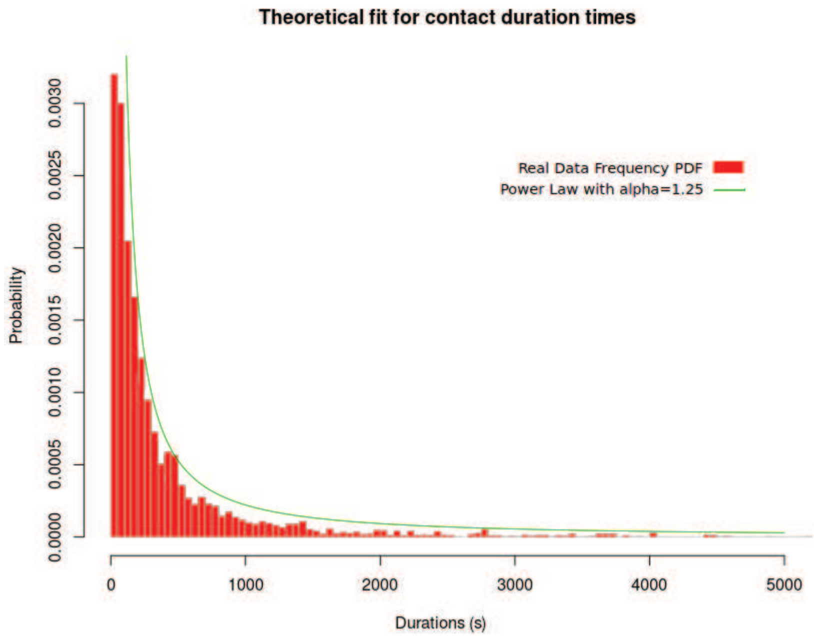


Figure 5.6: Fitting Contact Duration times by two theoretical distributions.

5.2.5 Inter-Contact durations

The same assumption made in Section 5.2.4 on the preceding page could be adopted for *inter-contact durations*, but only when the users are in the school. When a user is not in the school the *On-Off* distribution is not applicable because laptops are used but there are no contacts, so the *inter-contact time* distribution will be used in order to determine the time that a node will remain at home until it returns to school. This means when the laptop returns to school and not when the child returns to school, because he could go to school without his laptop and in this case the *inter-contact time* will be grater that one day.

The distribution of *inter-contact times* was modeled by a simple linear interpolation method using the real data itself. Figure 5.7 shows the estimated frequency histogram and the cumulative density function of the random numbers generated, showing the similarity with the real data. Figure 5.8 present the comparison of the real data distribution against the empirical distribution of data created by the generator, showing that the real information gathered from school is well fitted.

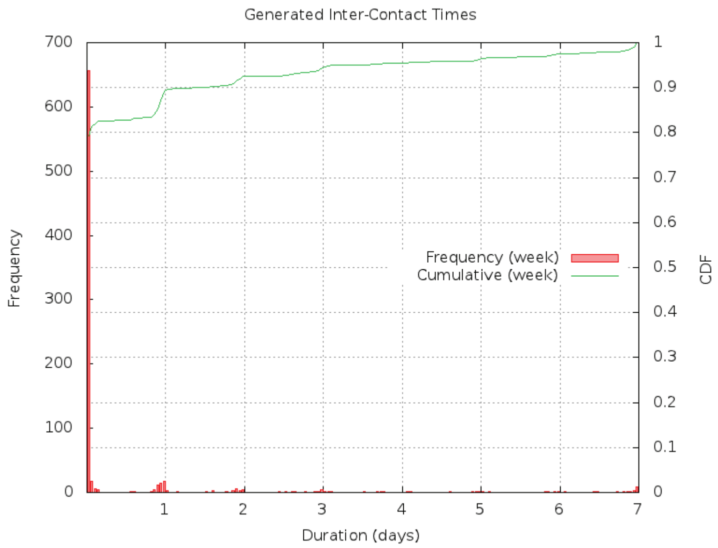


Figure 5.7: Estimated distribution of Inter-Contact times.

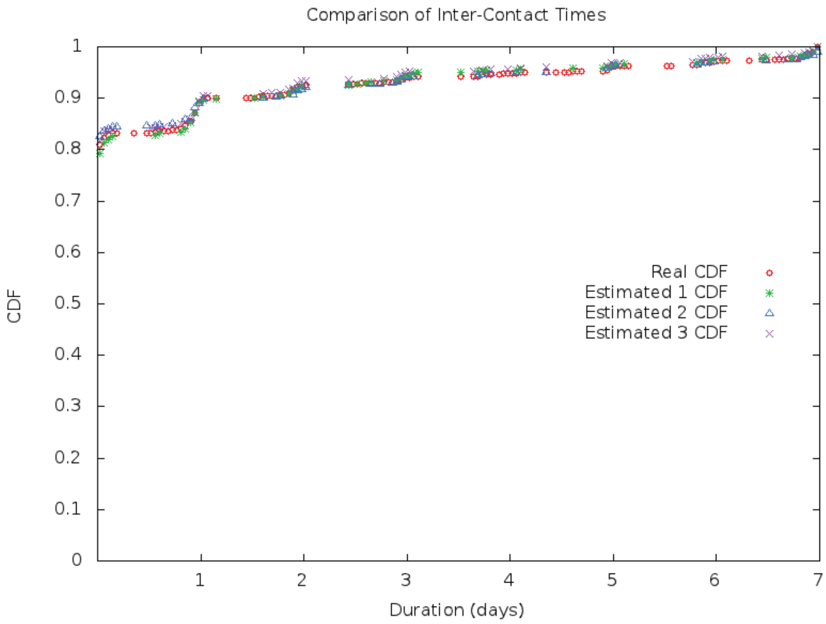


Figure 5.8: Comparing real distribution vs built random generator.

Chapter 6

Evaluation

This section performs the validation of the proposed model by comparing the metric values for generated traces against the real data.

6.1 Objectives

The main objective is to determine if the connectivity information derived from the real traces are well modeled by the proposed Demos Connectivity Model. The comparison is made using the same metrics derived from the real traces, and checking that the statistical values generated by the generator tool are equivalent.

In addition, we also prove that mobility traces generated by the model could be loaded into *ns3* and in particular demonstrate that the generated *On-Off* events can also be included in the simulation of the applications developed for *DEMOS*. Since *On-Off* property was identified as extremely important, simulations are performed to compare the differences in the results that use this feature and contrasting it with scenarios in which nodes remain on for the entire simulation.

6.2 Comparison of connectivity information

The comparison of connectivity information is made by comparing the real data vs the generated data using the metrics of: *on-off*, *contact durations* and *inter-contact durations*.

6.2.1 On-Off

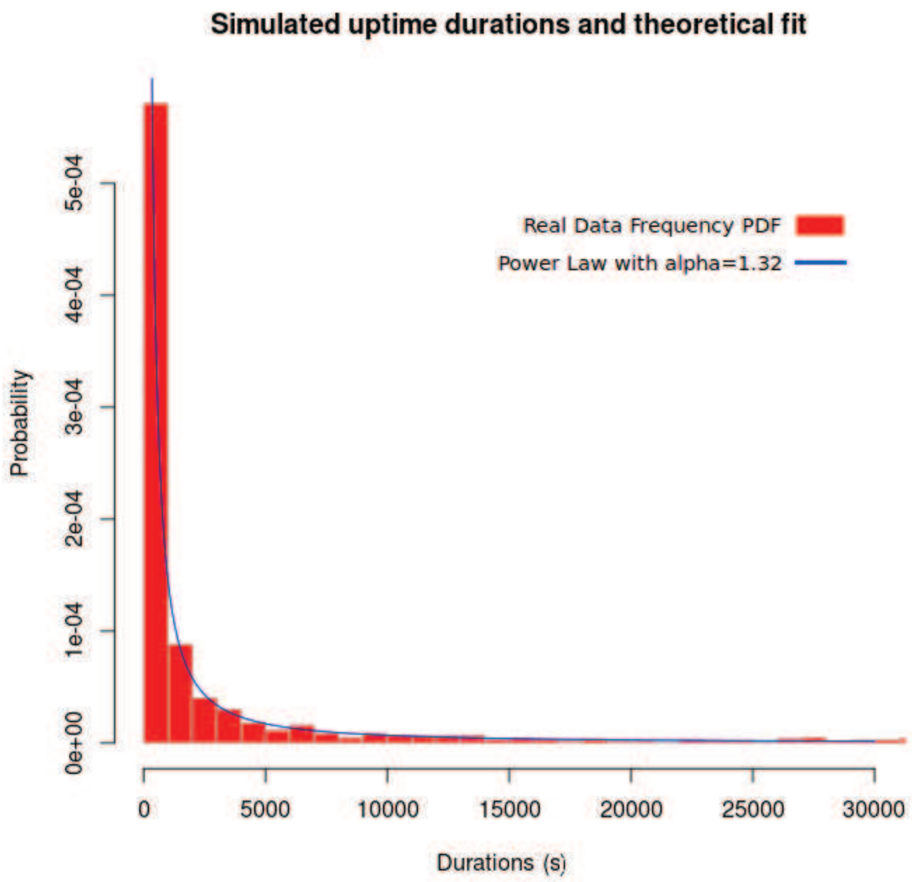
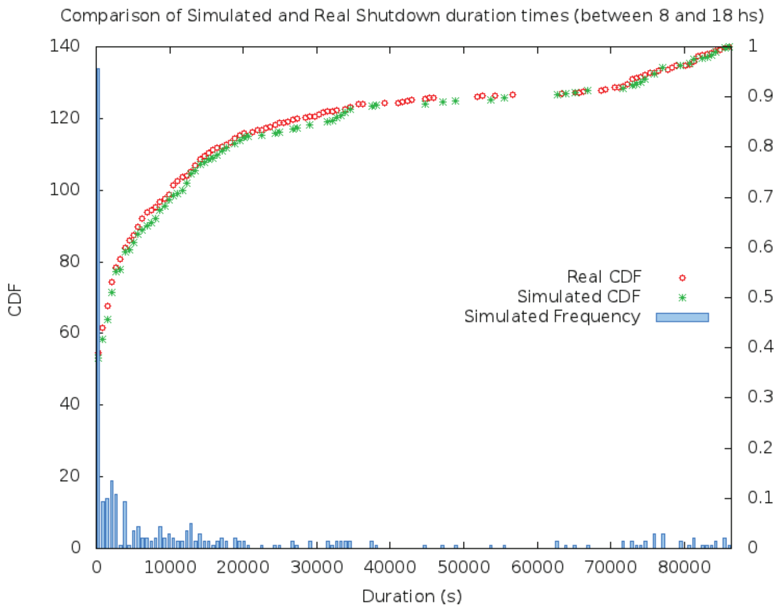


Figure 6.1: Fit of the uptime duration times by a Power Law theoretical distribution.

On-Off distributions were characterized in 5.2.3 and *DCM* makes use of those distributions directly to the generate duration times of each on and off event during the simulation. In this sense, we can expect a perfect match between the duration times generated during the generation of traces and the distributions that models the real data as shown in Figures 6.1 and 6.2.



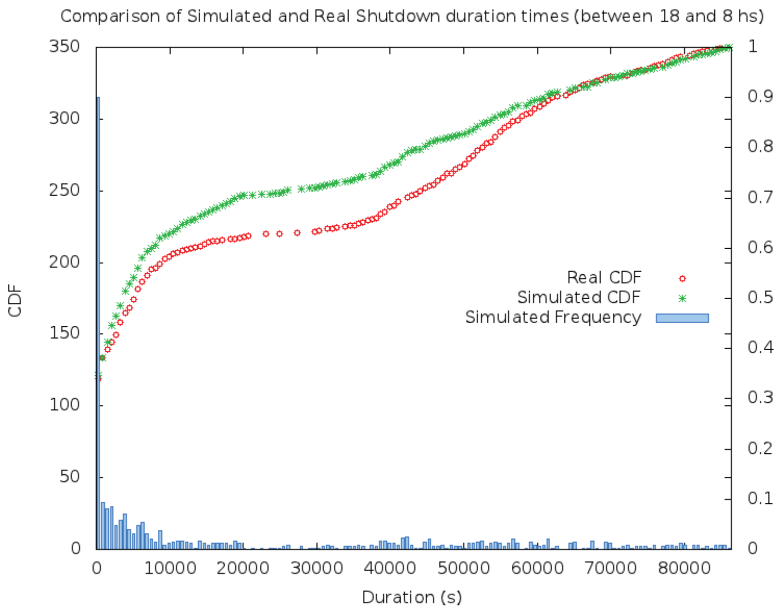


Figure 6.2: Fit of the shutdown duration times by hours.

6.2.2 Contact durations

Contact Duration is a derivative metric from the uptime duration times, and in the model a contact between nodes takes place when two nodes are both in the school and started on in a time frame. Real traces have shown that uptime and contact durations are both characterized by the same power law distribution, and this result is confirmed by 6.3.

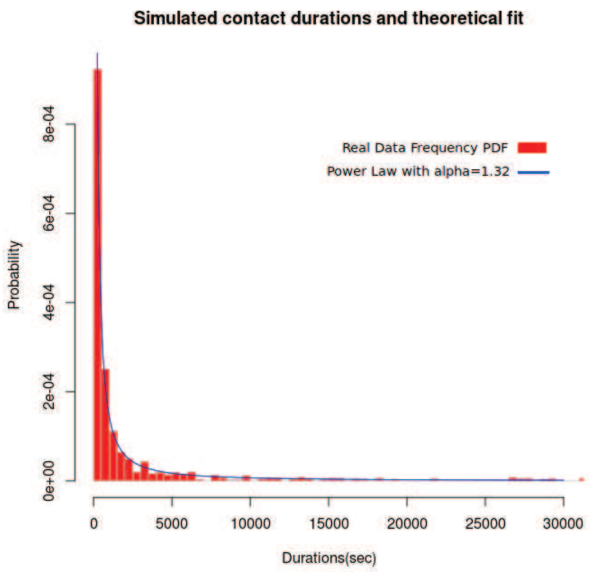


Figure 6.3: Fit of the contact time duration by a power law.

6.2.3 Inter-Contact durations

As in the case of *contact durations*, *inter-contact* times are derived from the uptime, shutdown times, and the position of nodes. An interesting point shown by this metric is that most of the times are minor than one day, and the nodes presents a periodicity, that in the cumulative density graph is represented by steps around integer values on the x-axis (days). Figure 6.4 on page 88 shows the real *CDF* vs distinct runs of the model for the *ICT* metric.

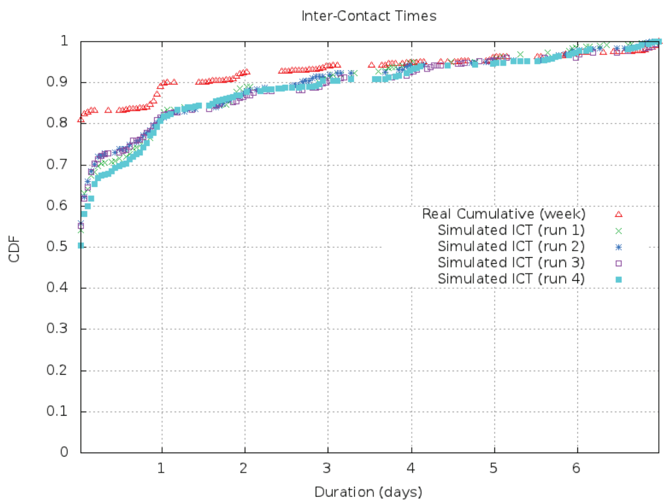


Figure 6.4: Fit of the inter-contact time duration against the real data.

6.2.4 Changing number of nodes

A very interesting result is that the model permits to change the number of nodes, and to generate connectivity traces that maintains the same *contact duration* and *inter-contact time* distributions (see Figures 6.5 and 6.6). This result permits the testing of different scenarios, changing the number of nodes but with the same patterns of connectivity, converting the *DCM* in a kind of synthetic parametric model. The four configured scenarios simulated are identical, except for the chosen number of nodes 50, 200, 300 and 500 nodes.

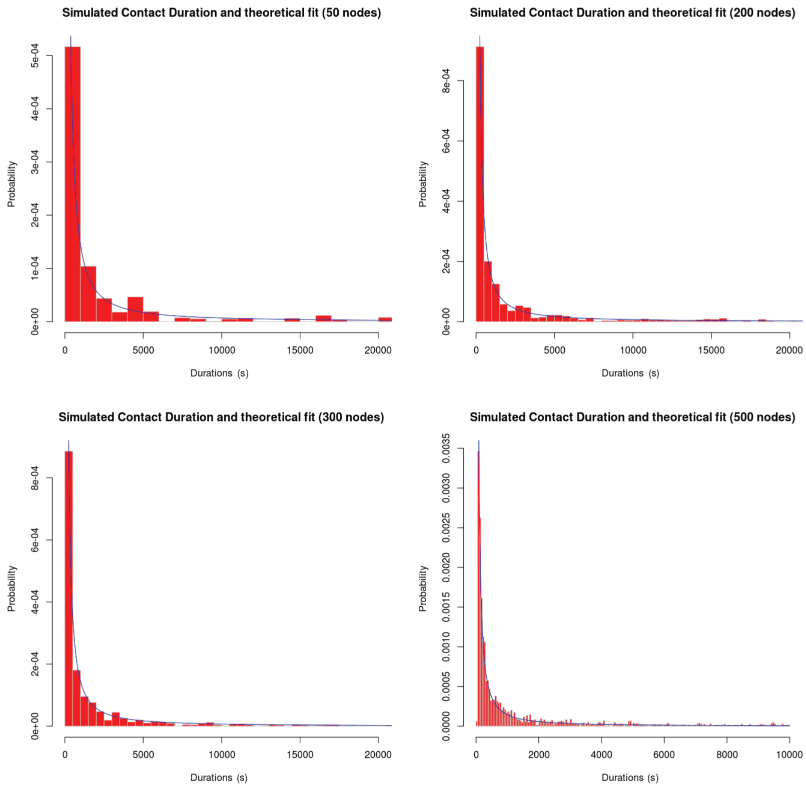


Figure 6.5: Fit of contact duration for distinct number of nodes by the same power law with alpha=1.32.

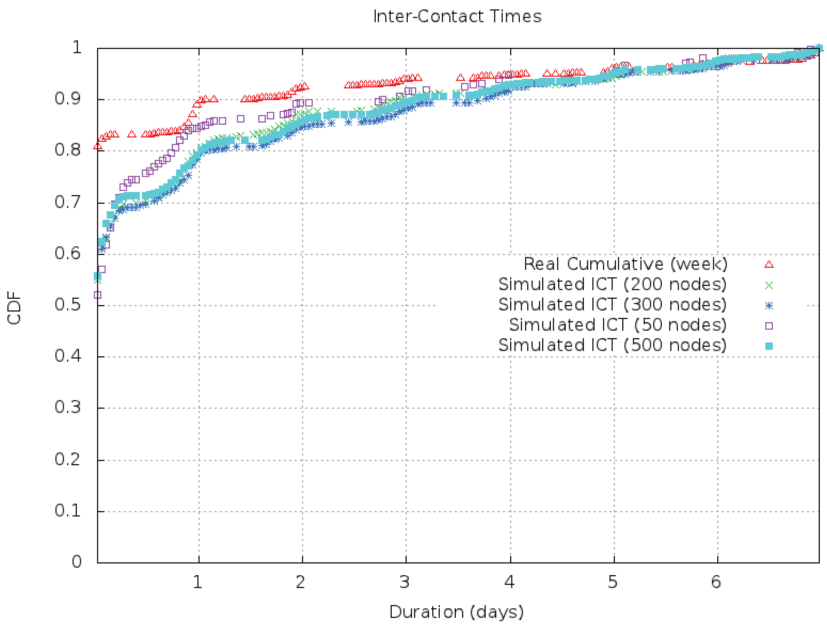


Figure 6.6: ICT distribution for different number of nodes.

6.3 Generated traces

The *DCM* software tool generates for each run a file containing mobility traces that reflects the same statistical behavior derived from the real traces. Traces contains information about node position and events for starts and shutdowns of each one, and these properties are enough to generate the same distributions of *contact durations* and in *inter-contact times* as shown in the previous section.

The generated mobility traces are structured as follow:

- The first lines of the traces are used to set the position of nodes in their homes at the beginning of the simulation, and to set the state of the node as turned off.

```
$node_(0)   set X_    50.0
$node_(0)  set Y_    50.0
$node_(0)   set Z_    0.0
$ns_ at 0 $node_(0) shutdown
$node_(1)   set X_    150.0
$node_(1)   set Y_    50.0
$node_(1)   set Z_    0.0
$ns_ at 0 $node_(1) shutdown
..........
```

- After that, the simulation begins, and the events of each node are the startup and shutdown of nodes and the movement of a node from his home to school and back. The generation of movements takes in consideration the weekends, and for these days, the movement of a node from home to school or vice versa are not computed. An example of traces generated for a node is as follow (the number in the left of the lines does not belong to the generated file, it is only for reference):

```
1  $ns_ at 43337.0 $node_(87) startup
2  $ns_ at 43337.0 "$node_(87) setdest 150.0 250.0 86023.25267042627"
3  $ns_ at 43409.0 $node_(87) shutdown
4  $ns_ at 47581.0 $node_(87) startup
5  $ns_ at 47776.0 $node_(87) shutdown
6  $ns_ at 77056.0 $node_(87) startup
7  $ns_ at 47776.0 "$node_(87) setdest 8750.0 50.0 86023.25267042627"
8  $ns_ at 93553.0 $node_(87) shutdown
9  $ns_ at 93623.0 $node_(87) startup
10 $ns_ at 94040.0 $node_(87) shutdown
11 $ns_ at 94117.0 $node_(87) startup
12 $ns_ at 94491.0 $node_(87) shutdown
13 $ns_ at 108753.0 $node_(87) startup
14 $ns_ at 114010.0 $node_(87) shutdown
15 $ns_ at 169130.0 $node_(87) startup
16 $ns_ at 169281.0 $node_(87) shutdown
17 $ns_ at 171477.0 $node_(87) startup
18 $ns_ at 171596.0 $node_(87) shutdown
19 $ns_ at 186221.0 $node_(87) startup
20 $ns_ at 186645.0 $node_(87) shutdown
21 $ns_ at 186657.0 $node_(87) startup
22 $ns_ at 222581.0 $node_(87) shutdown
23 $ns_ at 222581.0 "$node_(87) setdest 150.0 250.0 86023.25267042627"
24 $ns_ at 224094.0 $node_(87) startup
25 $ns_ at 259370.0 $node_(87) shutdown
26 $ns_ at 224094.0 "$node_(87) setdest 8750.0 50.0 86023.25267042627"
```

- Lines 2 and 23 represent that the node moves from home to school, and lines 7 and 26 represent movements from school to home. The rest of the lines define the times at which each node turns on and off.
- The syntax of the traces are similar to the syntax of the *ns2* network simulator:
  - Lines describing movements are valid sentences of *ns2* that instructs to the simulator that at the specified time, the mentioned node must start to move to the specified destination (defined by its (x,y) coordinates in the simulation area).
  - Lines describing the startup and shutdown events for a node are not valid in *ns2*, because those methods are not presents in the simulator, but the rest of the line is syntactically valid.

6.4 Simulations

Some previous results in *DEMOS* were performed using *DMM [44]* as mobility model that includes properties of daily periodicity and meetings at school every day, but it does not reflect realistic time distributions and weekends nor allows nodes on and off during the simulations. Thus, when a notification generated by a sensor that is located close to the child's home, the same could be received by the node that models the *XO* laptop immediately at best, and with a delay of approximately 7 hours at worst case. Moreover, once a notification is issued by a sensor and obtained

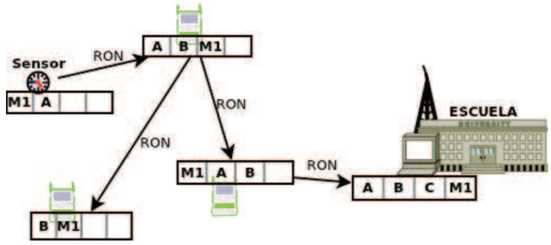


Figure 6.7: Internal *DEMOS* node buffers.

by the node, and assuming a best case in which notification is propagated to the server without being lost by buffer congestion of the opportunistic protocol, it can be say that the notification will arrive in less than 24 hours, since by construction of the model, child attend school every day. To compare the new results obtained from the use of *DCM* in contrast to previous results, qualitative analysis were performed on these, mainly measuring trends instead of trying to get exact values of certain performance values. A series of simulations were conducted using the configuration values summarized in Table 6.1 on page 93.

6.4.1 Delay time from movement pattern

The first experiment measures the delay of notifications to reach the central server, using the movement pattern generated with *DCM* but not including on-off events, in order to compare directly with the delay times that were visible with *DMM*. This result will allow us to make a comparison of the delay time with the sole effect of changing the pattern of mobility. Figure 6.8 on page 94 shows the existence of notifications with delay times greater than 24 hours, contrary to what happens with *DMM*. This is an important fact that has a strong impact on the parameters that measure the sensors, since notifications may take a long time to be processed by the central server. This delay reflects two factors that are not represented in *DMM* but which are modeled by *DCM*, namely: (1) the concept of weekdays and weekends are present in *DCM*, then it may exists notifications with delay time greater than 48 hours (this is not present in this simulation since it lasts for only 3 week days), and (2) even in school days, *DCM* models the fact that some laptops do not travel to school every day, or even when they travel, they still remain off during class schedule.

6.4.2 Impact of On-Off events in delay time

This section goes a step further, and uses the *DCM* model including *On-Off* events for the nodes. Since *ns3* does not include this functionality, it has been implemented as part of the *DEMOS* software, based on the times generated by *DCM* for each node. When *DCM* indicates that a node should be turned on, the software enters its normal operation, but when *DCM* indicates that a node should be turned off, the opportunistic protocol begins to discard all notifications that come and stops sending notifications already present in its buffer, keeping it intact until a new power on event occurs. Here is shown the importance of modeling off time of nodes in the delay of notifications by means of simulations performed using the same movement pattern, but keeping nodes always on in one case and including the on-off events for nodes in the other.

Figure 6.9 on page 95 compares the elapsed time between the generation of a notification by the sensor and when it is received by the corresponding node. This comparison is shown for two

PARAMETER	***PARAMETER VALUE***
3 week days	A mobility scenario was generated by *DCM* and the simulated time corresponds to 3 week days. While it is true that for a deep study of a particular application should be used a simulation time longer than one week (7 days), for a qualitative study to show the relevance of certain model parameters 3 days are sufficient. Qualitatively, including weekend days implies that the delay times are higher.
114 nodes	*XO* laptops that represent the same number of nodes used in the real network where the connectivity data were obtained.
buffer size of 200 messages	A buffer size of 200 messages is used on each node for the first set of simulations. For the last set of simulations the buffer size was configured with the values 20, 50, 100, 150 and 200 messages.
2 days timeout	A 2 days timeout for a notification to be removed from the buffer, set in relation to the total simulation time, which was set in 3 days. Since *DCM* includes the fact that a child miss school one day, this value would negatively impact the amount of notifications that reach the central server, as there will be a few whose time to live expire before being delivered to the school.
10 sensors generating one new notification per hour.	This value is taken empirically, with the aim of generating a reasonable number of notifications that have statistical significance, and that will avoid more complex problems in the system generated when there are many interacting notifications between many nodes.
1 central server	A node that represents the central server which collects notifications is used.
message buffers on each node fully occupied	Each node in the network maintains an internal buffer, where the notifications are stored while they are valid (see Figure 6.7 on page 92). For consistency, the simulation begins with each buffer's node fully occupied, so result data can be taken immediately assuming the system is in steady state.

Table 6.1: Configuration values of the simulated scenarios.

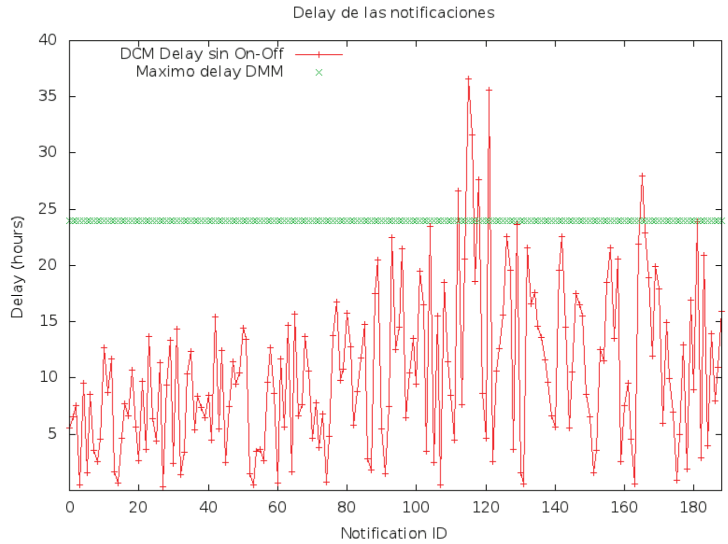
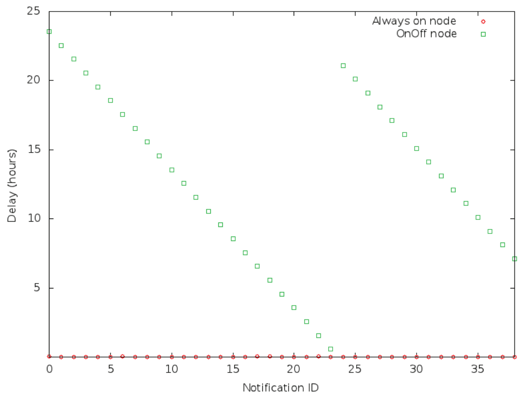


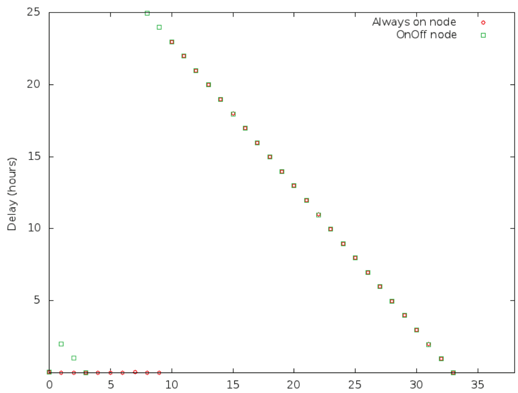
Figure 6.8: Delay notification times using DCM without on-off.

particular nodes as an example, but it is representative of the behavior for the other cases, and confirms that it is properly simulating the nodes on and off. In Figure 6.9a on page 95 we have a sensor that is close to the child's home and the laptop stays on at home all the time, and as expected the time taken for a node to receive notifications is in the range of a few seconds after the time they were generated by a sensor. On the other hand in Figure 6.9b on page 95 the same motion pattern was used but including on-off events, then although the node is close to the sensor, some notifications takes hours to reach the node because it stood off for that period (like notifications with ID from 0 to 2 and from 4 to 10). For the rest of notifications the delay to reach the node are the same because the node was not at home, and when the node returns hours later, it gets all the notifications previously generated by the sensor. Note that in Figure 6.9a on page 95 the node goes to school after 9 hs and it delivers notifications from 0 to 9 to the central server, but in Figure 6.9b on page 95, when the laptop arrives to school, it only propagates notifications from 0 to 2, because the notifications generated between 3 and 9 A.M. could not be obtained because XO was switched off during that time.

Once we have seen the impact of on-off events on the transit time after a notice is generated in the sensor and until it is collected by the corresponding node, we will see the effect on the end-to-end delay time, that is, since the notification is generated at the sensor until it is collected by the central server at school. The comparison shown in Figure 6.10 on page 96 was performed using the same movement pattern as in the above simulations, maintaining the nodes turned on all the time in one of them, while on-off events were included in the other. The figure shows the difference between the time when each notification reached the server (4 nodes were chosen representing different possible scenarios). In this sense, always-on nodes usually make notifications take less time to reach the central server, partly because scenarios using on-off events often get significant delays for a notification to reach its corresponding node. We also see delays that are much longer than 24 hours (the maximum reported delay value for *DMM*), which implies that the time of notification delivery should be taken into account in the case of applying the protocol in a real network.



(a) Delay for two nodes: one always on in its house, and the other with mobility pattern and always on too.



(b) Delay for two nodes: one always on in its house, and the other with mobility pattern and on-off.

Figure 6.9: Delay notification time from sensor to node.

6.4.3 DEMOS performance

As a last point of the evaluation, a study of a specific scenario that includes on-off events was made, trying to measure delay and loss rate of notifications. This test has a limited scope to verify that the generated traces can be used with the network simulator *ns3*, and particularly, that it is possible to perform simulations using the on-off feature, showing that it is possible to simulate a realistic and representative network for the *DEMOS*. The experiment measures accordingly the applicability of the protocol in a certain scenario, allowing to decide whether these parameters are acceptable or not within the reality that we want to monitor. The scenario was simulated by varying the size of the internal buffer that uses the *DEMOS* opportunistic protocol with the values 20, 50, 100, 150, and 200, trying to measure the impact of this parameter (among many configuration parameters that have the protocol) on delay and loss rate. The simulation was performed using 10 sensors, a central server located at the school and 114 nodes (XO laptops) for a period of 3 days. Sensor nodes will always issue data in a regular manner, with a certain given frequency (one notification per hour).

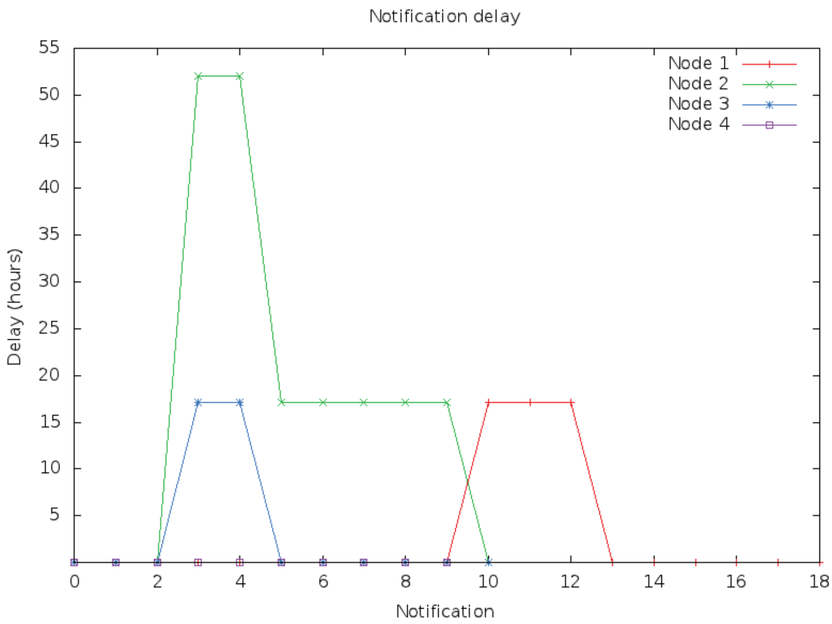


Figure 6.10: End-to-end delay time.

Figure 6.11 on page 97 shows that the number of notifications arriving to the central server increases with the size of the buffer as well as the delay increases. Using small buffer sizes favors that notifications reach the central server in less time, but in contrast the number of notifications that effectively arrive to server is lower. The number of notifications that arrive to the central server can vary because, as already mentioned, the opportunistic protocol has an internal finite buffer to store the notifications that when filled, the protocol must decide which notification of the buffer will be replaced (using a FIFO policy). Maybe a replaced notification has not yet reached the central server and if it was the only copy in the network, in that case it could be said that that notification will never reach its destination (the notification is lost). From Figure 6.11 on page 97 we can say that while the notifications reach their destination faster, these scenarios have a higher loss rate than those using larger buffers. In short, the buffer size should be selected to offer a compromise between the importance of having few delay notifications, or if instead we need a high reliability for notifications to be delivered to the central server.

Up to now we have considered for comparison only those notifications that reach the central server, but we missed a very important fact that is referring to the loss rate. This metric is really important, since in certain cases it is preferable that notifications reach the destination, rather that they arrive quickly. That is why Figure 6.12 on page 98 shows a comparison of the metrics explained below:

- The *absolute rate of arrival notification* refers to the ratio between the number of notifications generated by the sensor, and those that were ultimately received by the central server. This is called absolute rate because the calculation does not take into consideration any additional aspect concerning the model. As shown in the graph, this value is enhanced as we increase the buffer size, distinguishing a substantial improvement from size 50 in contrast to a size 20, and then, minor improvements are appreciated as it continues to increase the buffer size.

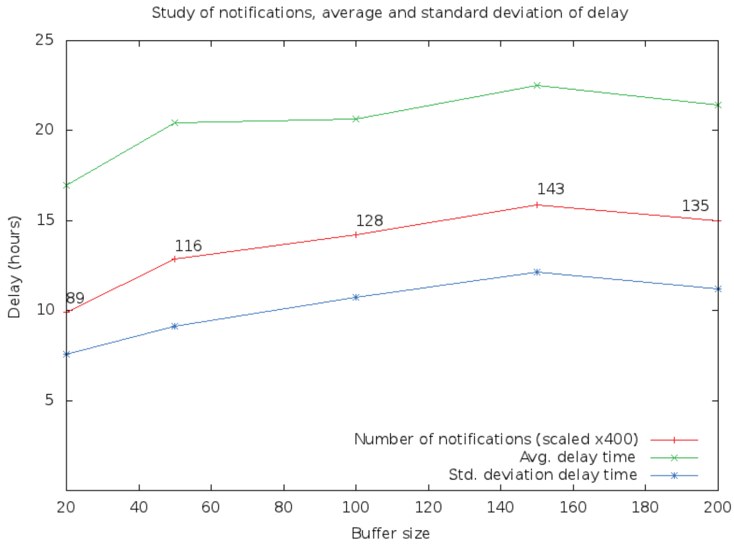


Figure 6.11: Number of notifications and delay from sensor to central server.

- *Arrival rate of notifications propagated by sensors* refers to the ratio between the number of notifications generated by sensors that were successfully received by their corresponding nodes, and the number of notifications that finally arrived to the server. Note that this number differs from *absolute rate of arrival notification* mainly because the sensor node running *DEMOS* software losses notifications due to lack of buffer space, so it could be said that a lost notification is a notification that failed to enter in the opportunistic network. When the ratio is calculated against the effective notifications instead of using all generated ones, arrival rate increases. As shown in Figure 6.12 on page 98, this metric increases significantly for a buffer size of 20 and the rate messages lost by the sensor actually increases too (in case of a small buffer will have many replacements). Notification loss may also have an origin related to on-off events and lifetime that *DEMOS* software keeps a notification before being removed from the buffer. Since a node can be close to a sensor, but off, notifications present in the sensor are not propagated and maybe by the time the child turn on the laptop within the sensor range, the notification lifetime may be expired.

- *Arrival rate of notifications without off times* calculates the ratio between the number of notifications generated by the sensor, and those that were ultimately received by the central server. This metric does not take inconsideration notifications that corresponds to nodes that have not attended school or notifications that by the times in which they were generated and the time when the node is kept on is impossible to have reached the server when the simulation ends. While it is true that in the notification did not reach its destination, this actually seems to be improved by means of configuration of the opportunistic protocol, increasing the simulation time and/or increasing the lifetime of each notification within the buffer or the buffer size. Of course, these settings may have an impact, for example in the delay of notifications.

Eventually, the loss rate shown in Figure 6.12 on page 98 quantifies the number of notifications that do not reach the corresponding node from the sensor. There is a great difference for the case of buffer size 20, where the two causes for the high rate of notification loss are met: frequently buffer replacements, or nodes are not powered on when they are close to the sensor while the notification is available.

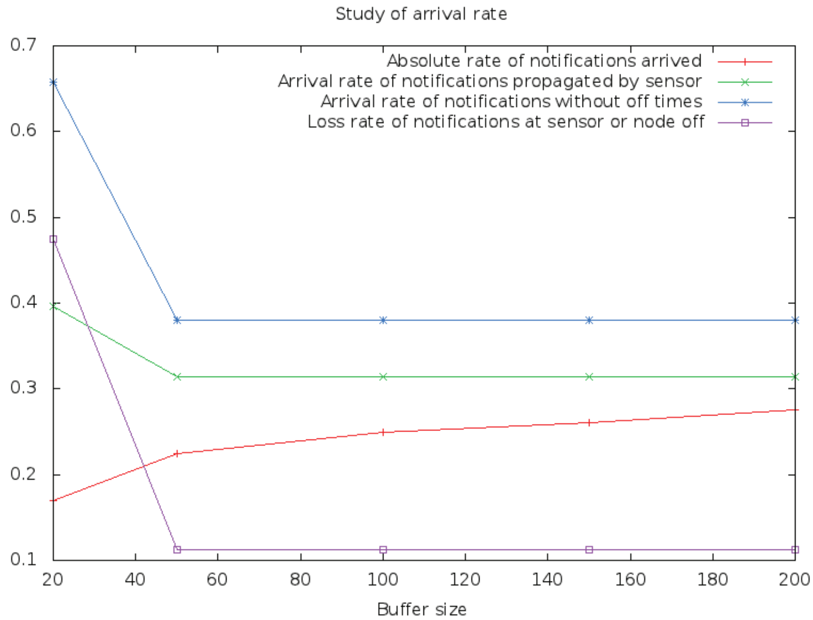


Figure 6.12: Arrival and loss rate of notifications.

6.5 Summary

The experimental evaluation has shown that the connectivity features derived from the real trace data are correctly captured by the proposed model. In this sense we can conclude that the model is able to generate accurate traces for the scenario under study. Furthermore it has been shown that traces of movement of the proposed model could be loaded in the network simulator *ns3* and the on-off events have also been included in the simulations with a small modification of the *DEMOS* software. Through simulations it has been proven how affects results when on-off events are taken into account, and a preliminary study of the performance of the protocol was performed, varying the buffer size of the *DEMOS* protocol and showing the impact on the rate of arrival, loss, and delay of notifications. Finally we can ensure that in such opportunistic scenarios, the mobility model and the use of on-off events is critical to obtain accurate results.

The output of the model generator is a file of movements and patterns of on-off events, which is able to instruct a network simulator to generate network scenarios similar to the network under study. Even more, the model permits to change the number of nodes, and generates connectivity traces that maintain the same *contact duration* and *inter-contact time* distributions, converting to the *DCM* a kind of synthetic model, because it can be used to generate different scenarios from the same set of initial traces.

Chapter 7

Conclusions and future work

This section presents the main conclusions of this thesis, proposes guidelines to follow as future work, and also suggestions to implement the model in real *Plan Ceibal* networks.

7.1 Conclusions

Mobility modeling is a key aspect for network simulations, and it has a great impact on the performance and accuracy of results. Literature about mobility models is wide but at the same time disperse, and the present work reviews exhaustively the state of the art in mobility modeling, by describing many models individually, being a good start point for the mobility modeling study. Given the large number of models, it is important to know the details of reality under study before we can choose the right model to use. In general, it should be noted that the models can be classified into categories according to their characteristics, and these categories allow to group models, from the simplest and inaccurately, to those that reflect reality adequately but that they are not always possible to use because they are created starting from real traces that in many cases are not available. It is clear that most of the synthetic models lack real behavior and that is why the *trace-to-model* approach is preferred for some scenarios, even when the task of getting real traces is more difficult.

Currently, we see a significant increase in the creation of models based on social behavior, responding to the increase of mobile devices and applications, trying to model a more realistic movements. In addition, there is a new branch in the study of mobility that is focused on the characterization of the connectivity behavior instead of studying in detail the location of individual entities. Connectivity models emerge as a viable alternative for the study of mobility in the area of opportunistic networks. Studies of social theory have reported interesting and general results on the mobility behavior of people, allowing the connectivity models to use that knowledge to reflect them on their construction.

The main result of this thesis is proposal of the *DCM* model, which is in essence a connectivity model, but it takes a step further, including explicitly *On-Off* events for nodes and positions in the simulation area that could be used in real network simulators. Through simulations, the importance of the *On-Off* property could be observed, suggesting that it is a relevant feature to be included in mobility scenarios and proving that maintaining nodes on during all the simulation time may yield inaccurate results. *DCM* was evaluated against real traces, showing to behave similar for the relevant connectivity metrics used. In addition, the document describes a method for data collection and a methodology to derive the main connectivity distributions. The process to get real traces has proved to work accurately in a real environment for devices with limited resources, and the tools for extraction of connectivity distributions were suitable for the this environment.

During the process of the present work, a few contributions are made:

- an intuitive mobility model for *Plan Ceibal* networks was proposed [44].
- contribution to the *ns3* simulator, developing a new mobility helper module to able to load *ns2* mobility traces in *ns3* [129].
- contribution to the *BonnMotion* [3]documentation, and indirectly by [129], allowing to load traces generated by *BonnMotion* (and others trace generators).

The study of mobility in *Plan Ceibal* network environments, beyond the theoretical results of the models themselves, has as a consequence the possible application to solve real-life situations based on that information. Some of these applications are:

- The main use of the model is to use it in a simulator, and at the same time running the applications (or trace generators for them) to study the performance inside the school building and outside it.
- From the use of XO laptops in class or at home: with information of On-Off patterns it could be possible to know in a statistical way how much the children are using their laptops. If this added information shows strange tendencies, a deeper analysis could be conducted to get to the root causes, that maybe are indicating that laptops are not being used in class, that many computers are not working properly, that children do not use laptops at home, or other causes.
- From a deep generalization study on the mobility (as mentioned in future work), a system to detect non standard behavior of children could be obtained. Some of this information could be used in conjunction with the anti-theft security system [110] to try a proactive approach.
- With the addition of some basic information, for example a list of applications that the user works with, a profile of the most used applications can be conducted, and this information can reflect the time the users spend working on curricular topics.

7.2 Future work

It seems to be inadequate to generalize the results obtained from the analysis of the real traces to all schools (e.g. rural schools). It could be necessary to perform the experimental evaluation for a set of schools with different characteristics in order to obtain good approximations for each metrics, that enable the creation of a school categorization grouped by their connectivity patterns. In the same way of generalization, a study of large geographical areas should be done, including more than just one school, to understand in a good way the behavior of a wide opportunistic network.

Even when the model permits to change the number of nodes maintaining the statistical distributions of the main connectivity metrics like *contact duration* and *inter-contact times*, this result does not have a real relevance until a study on several schools has been made. This feature is a very useful one and could be used in laboratory environment in the same way that the rest of synthetic model works, but bearing in mind that this result was not validated against real data.

The use of the movement trace lines (not the *startup* and *shutdown*) included in the traces generated by the tool could be loaded in *ns2* and in *ns3* using [129] but to make use of on-off events in a network simulator the simulator must support this functionality. Support of this functionality is not available yet in *ns3* but it is identified as an important improvement and is an open topic of discussion. The implementation of this feature could be another important contribution from *DEMOS* to *ns3*, not only because it is important for this work, but also because it would also be a significant contribution to the research community.

List of Figures

List of Tables

Glossary

Ad-Hoc A point to pint wireless connection between peers

AI Artificial Intelligence

AP Access Point

CD Contact Duration

CDF Cumulative Density Function

FIFO First In First Out

GPS Global Positioning System

Gravity Model In some wireless communication systems, receivers may tend to move towards the signal source, looking for a better signal. For example, in a cellular system, if a user experiences a low quality of communication and can move around, he may try to move towards a Base Station.

Group Motion Model In ad hoc networks, communications are often among teams which tend to coordinate their movements (e.g., a remen rescue team in a disaster recovery situation). To support this kind of communications and movements, the Mobility Vector model can provide effcient and realistic group mobility models. Different group patterns can be represented using base vectors while deviation vectors show the individual behaviors of members in a group.

GSM Global System for Mobile Communications

ICT Inter Contact Time

Location Dependent Model Represents a collective mobility pattern in a specifc area. For example, if a node is on a freeway, its mobility vector has a common component which represent the direction and the allowed speed of the freeway. If we have a digitized map and traffic pattern based on the map, we can use the base vector to implement the collective mobility. When a node moves around the area, it acquires the location dependent base vector specified at the current position.

MANET Mobile Ad-Hoc NETwork

MN Mobile Node

MRDM Variation of the Random Direction model, called the Modified Random Direction model. In this model, nodes select a direction degree as before, but they may choose their destination anywhere along that direction of travel. They do not need to travel all the way to the boundary.

OLPC One Laptop Per Child project

PDA Personal Digital Assistant

Targeting Model Is a common pattern of mobility, where nodes move towards a target. Given the target coordinate, it is simple to calculate a proper base vector. When a node approaches a target, it reduces its velocity using negative acceleration factor and then pause when the mobility vector is adjusted to zero.

USB Universal Serial Bus

WLAN Wireless Local Area Network

DTN Delay Tolerant Network

Bibliography

[1] GloMoSim: A scalable network simulation environment [online]. Available: `http://pcl.cs.ucla.edu/projects/glomosim/`, (August 2012).

[2] L. A. Amaral, A. Scala, M. Barthelemy, and H. E. Stanley. Classes of small-world networks. *Proceedings of the National Academy of Science of U.S.A.*, 97:11149–11152, Oct 2000.

[3] N. Aschenbruck, R. Ernst, E. Gerhards-Padilla, and M. Schwamborn. BonnMotion: a mobility scenario generation and analysis tool. In *Proceedings of the 3rd International ICST Conference on Simulation Tools and Techniques*, SIMUTools '10, pages 51:1–51:10, Torremolinos, Malaga, Spain, 2010.

[4] N. Aschenbruck, E. Gerhards-Padilla, M. Gerharz, M. Frank, and P. Martini. Modelling mobility in disaster area scenarios. In *Proceedings of the 10th ACM Symposium on Modeling, analysis, and simulation of wireless and mobile systems*, MSWiM '07, pages 4–12, New York, NY, USA, 2007. ACM.

[5] F. Bai and A. Helmy. "Wireless ad hoc and sensor networks", Chapter 1: "A survey of mobility models", 2004.

[6] F. Bai, N. Sadagopan, and A. Helmy. IMPORTANT: a framework to systematically analyze the Impact of Mobility on Performance of Routing Protocols for Adhoc Networks. In *Proceedings INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*, volume 2, pages 825–835, 2003.

[7] J. Baliosian, J. Visca, M. Richart, G. Apollonia, L. Vidal, M. Giachino, and E. Grampin. Self-managed content-based routing for opportunistic networks. In *Integrated Network Management (IM), 2011 IFIP/IEEE International Symposium on*, pages 121 –128, May 2011.

[8] A. Barabasi and E. Bonabeau. Scale-free networks. *Scientific American*, 288(5):50–9, 2003.

[9] M. Behrisch, L. Bieker, J. Erdmann, and D. Krajzewicz. SUMO - Simulation of Urban MObility: An Overview. In *SIMUL 2011, The Third International Conference on Advances in System Simulation*, pages 63–68, Barcelona, Spain, 2011.

[10] A. Berl. Modeling Mobility. In K. Wehrle, M. Güneş, and J. Gross, editors, *Modeling and Tools for Network Simulation*, pages 327–339. Springer Berlin Heidelberg, 2010. DOI: 10.1007/978-3-642-12331-3_14.

[11] C. Bettstetter. Smooth is better than sharp: a random mobility model for simulation of wireless networks. In *Proceedings of the 4th ACM international workshop on Modeling, analysis and simulation of wireless and mobile systems*, MSWIM '01, pages 19–27, New York, NY, USA, 2001. ACM.

[12] C. Bettstetter, H. Hartenstein, and X. Pérez-Costa. Stochastic properties of the random waypoint mobility model. *Wireless Networks*, 10(5):555–567, September 2004.

[13] C. Bettstetter, G. Resta, and P. Santi. The Node Distribution of the Random Waypoint Mobility Model for Wireless Ad Hoc Networks. *IEEE Trans. Mobile Computing*, 2(3):257–269, July 2003.

[14] C. Bettstetter and C. Wagner. The Spatial Node Distribution of the Random Waypoint Mobility Model. In *Mobile Ad-Hoc Netzwerke, 1. deutscher Workshop über Mobile Ad-Hoc Netzwerke WMAN 2002*, pages 41–58. GI, 2002.

[15] D. Bhattacharjee, A. Rao, Shah C, M. Shah, and A. Helmy. Empirical modeling of campus-wide pedestrian mobility observations on the USC campus. In *Proceedings VTC2004-Fall Vehicular Technology Conference 2004 IEEE 60th*, volume 4, pages 2887–2891, 2004.

[16] 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.

[17] K. Blakely and B. Lowekamp. A structured group mobility model for the simulation of mobile ad hoc networks. In *Proceedings of the second international workshop on Mobility management & wireless access protocols*, MobiWac '04, pages 111–118, New York, NY, USA, 2004. ACM.

[18] L. Blažević, S. Giordano, and J. Le Boudec. Self Organized Terminode Routing. *Cluster Computing*, 5:205–218, April 2002.

[19] C. Boldrini, M. Conti, and A. Passarella. Users mobility models for opportunistic networks: the role of physical locations. In *IEEE Wireless Rural and Emergency Communications - WRECOM07*, pages 1–6, 2007.

[20] V. Borrel, F. Legendre, M. Dias de Amorim, and S. Fdida. SIMPS: Using Sociology for Personal Mobility. *Networking, IEEE/ACM Transactions on*, 17(3):831 –842, June 2009.

[21] S. Boschi, M. Di lanni, P. Crescenzi, G. Rossi, and P. Vocca. MOMOSE: a mobility model simulation environment for mobile wireless ad-hoc networks. In *Proceedings of the 1st International Conference on Simulation tools and techniques for communications, networks and systems & workshops*, Simutools '08, pages 38:1–38:10, ICST, Brussels, Belgium, Belgium, 2008.

[22] J. Broch, D. Maltz, D. Johnson, Y. Hu, and J. Jetcheva. A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols. In *Proceedings of the Fourth Annual ACM/IEEE International Conference on Mobile Computing and Networking (Mobicom98)*, pages 85–97. ACM, 1998.

[23] M. Buckland. *Programming Game AI by Example*. Wordware Publishing, 2005.

[24] R. Calegari, M. Musolesi, F. Raimondi, and C. Mascolo. A human connectivity model for opportunistic mobile systems. UCL CS Research Note RN/07/08.

[25] R. Calegari, M. Musolesi, F. Raimondi, and C. Mascolo. CTG: a connectivity trace generator for testing the performance of opportunistic mobile systems. In *Proceedings of the the 6th joint meeting of the European software engineering conference and the ACM SIGSOFT symposium*

on The foundations of software engineering, ESEC-FSE '07, pages 415–424, New York, NY, USA, 2007. ACM.

[26] 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.

[27] 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.

[28] A. Chaintreau, P. Hui, J. Crowcroft, C. Diot, R. Gass, and J. Scott. Pocket Switched Networks: Real-world mobility and its consequences for opportunistic forwarding. Technical Report UCAM-CL-TR-617, University of Cambridge, Computer Laboratory, February 2005.

[29] A. Chaintreau, P. Hui, J. Crowcroft, C. Diot, R. Gass, and J. Scott. Impact of Human Mobility on Opportunistic Forwarding Algorithms. *IEEE Transactions on Mobile Computing*, 6:606–620, June 2007.

[30] W. Chen and P. Chen. Group mobility management in wireless ad hoc networks. In *Proceedings VTC 2003-Fall Vehicular Technology Conference 2003 IEEE 58th*, volume 4, pages 2202–2206, 2003.

[31] CC. Chiang. *Wireless Network Multicasting.* PhD thesis, University of California, Los Angeles, 1998.

[32] A. Clauset, C. Shalizi, and M. E. J. Newman. Power-Law Distributions in Empirical Data. *SIAM Rev.*, 51(4):661–703, November 2009.

[33] Aaron Clauset. [Online]. Available: http://tuvalu.santafe.edu/ aaronc/powerlaws/. (November 2012).

[34] V. Davies. Evaluating Mobility Models Within An Ad Hoc Network. Master's thesis, Colorado School of Mines, Dept. of Mathematical and Computer Sciences, 2000.

[35] M. de Berg, O. Cheong, M. van Kreveld, and M. Overmars. *Computational Geometry: Algorithms and Applications.* Springer, 3rd edition, April 2008.

[36] E. W. Dijkstra. *A note on two problems in connection with graphs*, chapter 1, pages 269–271. Numerische Mathematik, 1959.

[37] F. Ekman, A. Keränen, J. Karvo, and J. Ott. Working day movement model. In *MobilityModels '08: Proceeding of the 1st ACM SIGMOBILE workshop on Mobility models*, pages 33–40, New York, NY, USA, 2008. ACM.

[38] D. Fischer, K. Herrmann, and K. Rothermel. GeSoMo - A general social mobility model for delay tolerant networks. In *Proceedings of the 7th IEEE International Conference on Mobile Adhoc and Sensor Systems (MASS)*, pages 99–108. IEEE, November 2010.

[39] A. Gainaru, C. Dobre, and V. Cristea. A Realistic Mobility Model Based on Social Networks for the Simulation of VANETs. In *Proceedings IEEE 69th Vehicular Technology Conference VTC Spring 2009*, pages 1–5, 2009.

[40] S. Gaito, G. Grossi, and F. Pedersini. A two-level social mobility model for trace generation. In *Proceedings of the 9th ACM international symposium on Mobile ad hoc networking and computing*, MobiHoc '08, pages 457–458, New York, NY, USA, 2008. ACM.

[41] L. Gavrilovska and R. Prasad. Mobility. In *Ad Hoc Networking Towards Seamless Communications*, Signals and Communication Technology, pages 173–209. Springer Netherlands, 2006. DOI: 10.1007/978-1-4020-5066-4_7.

[42] S. C. Geyik, E. Bulut, and B. K. Szymanski. PCFG Based Synthetic Mobility Trace Generation. In *Proceedings IEEE Global Telecommunications Conference GLOBECOM 2010*, pages 1–5, 2010.

[43] S. C. Geyik and B. K. Szymanski. Event Recognition in Sensor Networks by Means of Grammatical Inference. In *Proceedings IEEE INFOCOM 2009*, pages 900–908, 2009.

[44] M. Giachino, J. Baliosian, E. Grampín, and J. Visca. DEMOS Mobility Model. In *Proceedings 6th International Wireless and Mobile Communications (ICWMC) Conference*, pages 177–184, 2010.

[45] M. L. Goldstein, S. A. Morris, and G. G. Yen. Problems with fitting to the power-law distribution. *The European Physical Journal B - Condensed Matter and Complex Systems*, 41:255–258, 2004. 10.1140/epjb/e2004-00316-5.

[46] M. González, C. Hidalgo, and A. Barabási. Understanding individual human mobility patterns. *Nature*, 453(7196):779–782, Jun 2008.

[47] R. Groenevelt, E. Altman, and P. Nain. Relaying in mobile ad hoc networks: The Brownian motion mobility model. *Wireless Networks*, 12:561–571, 2006. DOI: 10.1007/s11276-006-6535-0.

[48] M. Gunes, F. Juraschek, B. Blywis, and C. Graff. MoNoTrac — A mobility trace generator based on OpenStreetMap geo-data. In *Proceedings IEEE 7th International Mobile Adhoc and Sensor Systems (MASS) Conference*, pages 618–623, 2010.

[49] M. Günes and J. Siekermann. CosMos - Communication Scenario and Mobility Scenario Generator for Mobile Ad-hoc Networks. In *Proceedings of the 2nd International Workshop on MANETs and Interoperability Issues (MANETII'05)*, pages 17–23, 2005.

[50] Z. J. Haas. A new routing protocol for the reconfigurable wireless networks. In *Proceedings IEEE 6th International Universal Personal Communications Record Conference*, volume 2, pages 562–566, 1997.

[51] K. Herrmann. Modeling the sociological aspects of mobility in ad hoc networks. In *Proceedings of the 6th ACM international workshop on Modeling analysis and simulation of wireless and mobile systems*, MSWIM '03, pages 128–129, New York, NY, USA, 2003. ACM.

[52] G.R. Hiertz, D. Denteneer, S. Max, R. Taori, J. Cardona, L. Berlemann, and B. Walke. IEEE 802.11s: The WLAN Mesh Standard. *Wireless Communications, IEEE*, 17(1):104 –111, February 2010.

[53] X. Hong, M. Gerla, G. Pei, and C. Chiang. A group mobility model for ad hoc wireless networks. In *Proceedings of the 2nd ACM international workshop on Modeling, analysis and simulation of wireless and mobile systems*, MSWiM '99, pages 53–60, New York, NY, USA, 1999. ACM.

[54] W. Hsu and A. Helmy. IMPACT: Investigation of Mobile-user Patterns Across University Campuses using WLAN Trace Analysis. *The Computing Research Repository*, abs/cs/0508009, 2005.

[55] W. Hsu and A. Helmy. On Modeling User Associations in Wireless LAN Traces on University Campuses. In *Proceedings 4th International Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks Symposium*, pages 1–9, 2006.

[56] W. Hsu, K. Merchant, H. Shu, C. Hsu, and A. Helmy. Weighted waypoint mobility model and its impact on ad hoc networks. *SIGMOBILE Mobile Computing and Communications Review*, 9:59–63, January 2005.

[57] W. Hsu, T. Spyropoulos, K. Psounis, and A. Helmy. Modeling Time-Variant User Mobility in Wireless Mobile Networks. In *Proceedings INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE*, pages 758–766, 2007.

[58] W. Hsu, T. Spyropoulos, K. Psounis, and A. Helmy. Modeling spatial and temporal dependencies of user mobility in wireless mobile networks. *IEEE/ACM Transactions on Networking*, 17:1564–1577, October 2009.

[59] R. Jain, D. Lelescu, and M. Balakrishnan. Model T: an empirical model for user registration patterns in a campus wireless LAN. In *Proceedings of the 11th annual international conference on Mobile computing and networking*, MobiCom '05, pages 170–184, New York, NY, USA, 2005. ACM.

[60] A. Jardosh, E. M. Belding-Royer, K. C. Almeroth, and S. Suri. Towards realistic mobility models for mobile ad hoc networks. In *Proceedings of the 9th annual international conference on Mobile computing and networking*, MobiCom '03, pages 217–229, New York, NY, USA, 2003. ACM.

[61] 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.

[62] P. Johansson, T. Larsson, N. Hedman, B. Mielczarek, and M. Degermark. Scenario-based Performance Analysis of Routing Protocols for Mobile Ad-hoc Networks. In *Proceedings of the Mobicom*, pages 195–206, 1999.

[63] T. Karagiannis, J. Le Boudec, and M. Vojnovic. Power Law and Exponential Decay of Intercontact Times between Mobile Devices. *IEEE Transactions on Mobile Computing*, 9(10):1377–1390, October 2010.

[64] T. Kathiravelu, A. Pears, and N. Ranasinghe. Connectivity Models : A New Approach to Modeling Contacts in Opportunistic Networks. In *Proceedings of the Eighth International Information Technology Conference (IITC) 2006*, pages 185–, 2006.

[65] A. Keränen, J. Ott, and T. Kärkkäinen. The ONE Simulator for DTN Protocol Evaluation. In *SIMUTools '09: Proceedings of the 2nd International Conference on Simulation Tools and Techniques*, New York, NY, USA, 2009. ICST.

[66] M. Kim, D. Kotz, and S. Kim. Extracting a Mobility Model from Real User Traces. In *INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings*, pages 1–13, April 2006.

[67] K. Konishi, K. Maeda, K. Sato, A. Yamasaki, H. Yamaguchi, K. Yasumoto, and T. Higashino. MobiREAL simulator-evaluating MANET applications in real environments. In *Proceedings 13th IEEE International Modeling, Analysis, and Simulation of Computer and Telecommunication Systems Symposium*, pages 499–502, 2005.

[68] D. Kotz and K. Essien. Analysis of a campus-wide wireless network. *Wireless Networks*, 11:115–133, January 2005.

[69] T. J. Kwon and M. Gerla. Clustering with power control. In *Proceedings IEEE Military Communications MILCOM 1999*, volume 2, pages 1424–1428, 1999.

[70] D. Lam, D. C. Cox, and J. Widom. Teletraffic modeling for personal communications services. 35(2):79–87, 1997.

[71] D. Lam, J. Jannink, D. Cox, and J. Widom. Modeling Location Management in Personal Communications Services. In *International Conference on Universal Personal Communications (ICUPC'96)*, volume http://ilpubs.stanford.edu:8090/173/, 1996.

[72] P. Lassila, E. Hyytiä, and H. Koskinen. Connectivity Properties of Random Waypoint Mobility Model for Ad Hoc Networks. In K. Al Agha, I. Guérin Lassous, and G. Pujolle, editors, *Challenges in Ad Hoc Networking*, volume 197 of *IFIP International Federation for Information Processing*, pages 159–168. Springer Boston, 2006.

[73] J.-Y. 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.

[74] 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.

[75] F. Legendre, V. Borrel, M. de Amorim, and S. Fdida. Modeling Mobility with Behavioral Rules: The Case of Incident and Emergency Situations. In K. Cho and P. Jacquet, editors, *Technologies for Advanced Heterogeneous Networks II*, volume 4311 of *Lecture Notes in Computer Science*, pages 186–205. Springer Berlin/Heidelberg, 2006.

[76] D. Lelescu, U. C. Kozat, R. Jain, and M. Balakrishnan. Model T++: an empirical joint space-time registration model. In *Proceedings of the 7th ACM international symposium on Mobile ad hoc networking and computing*, MobiHoc '06, pages 61–72, New York, NY, USA, 2006. ACM.

[77] B. Liang and Z.J. Haas. Predictive distance-based mobility management for PCS networks. In *INFOCOM '99. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 3, pages 1377–1384, March 1999.

[78] S. Lim, C. Yu, and C. R. Das. Clustered Mobility Model for Scale-Free Wireless Networks. In *Proceedings 31st IEEE Conference on Local Computer Networks*, pages 231–238, 2006.

[79] G. Lin, Noubir G, and R. Rajaraman. Mobility models for ad hoc network simulation. In *Proceedings INFOCOM 2004. Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies*, volume 1, 2004.

[80] X. Lu, Y. Chen, I. Leung, Z. Xiong, and P. Liò. A Novel Mobility Model from a Heterogeneous Military MANET Trace. In David Coudert, David Simplot-Ryl, and Ivan Stojmenovic, editors, *Ad-hoc, Mobile and Wireless Networks*, volume 5198 of *Lecture Notes in Computer Science*, pages 463–474. Springer Berlin/Heidelberg, 2008.

[81] Yenliang Lu, Huier Lin, Yajuan Gu, and A. Helmy. Towards mobility-rich analysis in ad hoc networks: using contraction, expansion and hybrid models. In *Proceedings IEEE International Communications Conference*, volume 7, pages 4346–4351, 2004.

[82] M. Bergamo, Hain R, K. Kasera, D. Li, R. Ramanathan, and M. Steenstrup. System design specification for mobile multimedia wireless network (MMWN) (Draft). Technical report, DARPA project DAAB07-95-C-D156, October 1996.

[83] K. Maeda, K. Sato, K. Konishi, A. Yamasaki, A. Uchiyama, H. Yamaguchi, K. Yasumoto, and T. Higashino. Getting urban pedestrian flow from simple observation: realistic mobility generation in wireless network simulation. In *Proceedings of the 8th ACM international symposium on Modeling, analysis and simulation of wireless and mobile systems*, MSWiM '05, pages 151–158, New York, NY, USA, 2005. ACM.

[84] K. Maeda, T. Umedu, H. Yamaguchi, K. Yasumoto, and T.H. Higashino. MobiREAL : Scenario Generation and Toolset for MANET Simulation with Realistic Node Mobility. In *Mobile Data Management, 2006. MDM 2006. 7th International Conference on*, page 55, May 2006.

[85] J. G. Markoulidakis, G. L. Lyberopoulos, D. F. Tsirkas, and E. D. Sykas. Mobility modeling in third-generation mobile telecommunications systems. *IEEE Personal Communications*, 4(4):41–56, 1997.

[86] M. McNett and G. M. Voelker. Access and mobility of wireless PDA users. *SIGMOBILE Mobile Computing and Communications Review*, 9:40–55, April 2005.

[87] A. Medina, G. Gursun, P. Basu, and I. Matta. On the Universal Generation of Mobility Models. In *Modeling, Analysis Simulation of Computer and Telecommunication Systems (MASCOTS), 2010 IEEE International Symposium on*, pages 444–446, August 2010.

[88] S. Milgram. The small world problem. *Psychology today*, 1(1):61–67, May 1967.

[89] Random Trip model [online]. Available: `http://icawww1.epfl.ch/RandomTrip/`, (August 2012).

[90] M. Sanchez. Mobility models [online]. Available: `http://www.disca.upv.es/misan/mobmodel.htm`, (August 2012).

[91] M. Musolesi, S. Hailes, and C. Mascolo. An ad hoc mobility model founded on social network theory. In *Proceedings of the 7th ACM international symposium on Modeling, analysis and simulation of wireless and mobile systems*, MSWiM '04, pages 20–24, New York, NY, USA, 2004. ACM.

[92] M. Musolesi and C. Mascolo. A community based mobility model for ad hoc network research. In *Proceedings of the 2nd international workshop on Multi-hop ad hoc networks: from theory to reality*, REALMAN '06, pages 31–38, New York, NY, USA, 2006. ACM.

[93] M. Musolesi and C. Mascolo. Designing mobility models based on social network theory. *SIGMOBILE Mobile Computing and Communications Review*, 11:59–70, July 2007.

[94] M. Musolesi and C. Mascolo. Mobility Models for Systems Evaluation. In B. Garbinato, H. Miranda, and L. Rodrigues, editors, *Middleware for Network Eccentric and Mobile Applications*, pages 43–62. Springer Berlin Heidelberg, 2009. DOI: 10.1007/978-3-540-89707-1_3.

[95] E. Gerhards-Padilla N. Aschenbruck and P. Martini. A survey on mobility models for performance analysis in tactical mobile networks. In *Journal of Telecommunications and Information Technology*, number 2/2008, pages 54–61, 2008.

[96] P. Nain, D. Towsley, B. Liu, and Z. Liu. Properties of random direction models. In *Proceedings IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies INFOCOM 2005*, volume 3, pages 1897–1907, 2005.

[97] W. Navidi and T. Camp. Stationary Distributions for the Random Waypoint Mobility Model. *IEEE Transactions on Mobile Computing*, 3(1):99–108, 2004.

[98] M. E. J. Newman. Assortative Mixing in Networks. *Physical Review Letters*, 89:208701, October 2002. DOI: 10.1103/PhysRevLett.89.208701.

[99] ns-3-dce [online]. Available: `http://code.nsnam.org/mathieu/ns-3-dce/file/38b25a74e8e7`, (August 2012).

[100] The Network Simulator ns2 Information Sciences Institute [online]. Available: `http://www.isi.edu/nsnam/ns/`, (August 2012).

[101] The Network Simulator ns3 [online]. Available: `http://www.nsnam.org/`, (August 2012).

[102] J. Nykvist and K. Phanse. Modeling connectivity in mobile ad-hoc network environments. In *Proceedings of the 6th Scandinavian workshop on Wireless Ad Hoc Networks(ADHOC 06), Johannesberg Slotts, Stockholm*, May 2006.

[103] Freie Universität Berlin DES-Workgroup MoNoTrac Project [online]. Available: `http://des-testbed.net/MoNoTrac`, (August 2012).

[104] Momose [online]. Available: `http://piluc.dsi.unifi.it/piluc/?page_id=140`, (August 2012).

[105] One Laptop Per Child [online]. Available: `http://laptop.org`, (August 2012).

[106] OpenStreetMaps Project [online]. Available: `http://www.openstreetmap.org/`, (August 2012).

[107] Plan Ceibal [online]. Available: `http://www.ceibal.edu.uy`, (August 2012).

[108] Stanford University Mobile Activity TRAces [online]. `http://infolab.stanford.edu/pleiades/SUMATRA.html`, (August 2012).

[109] open802.11s [online]. Available: `http://open80211s.org/open80211s/`, (August 2012).

[110] E. Pastorino and F. Haim. Anti-theft security system for Plan Ceibal XO laptop. In *Network Operations and Management Symposium, 2009. LANOMS 2009. Latin American*, pages 1–4, October 2009.

[111] 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, January 2008.

[112] P. Prabhakaran and R. Sankar. Impact of Realistic Mobility Models on Wireless Networks Performance. In *Wireless and Mobile Computing, Networking and Communications, 2006. (WiMob'2006). IEEE International Conference on*, pages 329–334, 2006.

[113] R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2012. ISBN 3-900051-07-0.

[114] T. S. Rappaport, editor. *Wireless Communications: Principles and Practice*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1995.

[115] M. Richart, J. Visca, J. Baliosian, and E. Grampin. A mobility model for opportunistic routing protocols validation. In *Network Operations and Management Symposium (LANOMS), 2011 7th Latin American*, pages 1–8, October 2011.

[116] R. R. Roy. *Handbook of Mobile Ad Hoc Networks for Mobility Models*. Springer US, 2011.

[117] R. R. Roy. Reference Velocity and Acceleration Group Mobility. In *Handbook of Mobile Ad Hoc Networks for Mobility Models*, pages 685–694. Springer US, 2011.

[118] R. R. Roy. Reference Velocity Group Mobility. In *Handbook of Mobile Ad Hoc Networks for Mobility Models*, pages 671–684. Springer US, 2011.

[119] E. M. Royer, P. M. Melliar-Smith, and L. E. Moser. An analysis of the optimum node density for ad hoc mobile networks. In *Proceedings IEEE International Conference on Communications ICC 2001*, volume 3, pages 857–861, 2001.

[120] C. Schindelhauer, T. Lukovszki, S. Rührup, and K. Volbert. Worst case mobility in ad hoc networks. In *Proceedings of the fifteenth annual ACM symposium on Parallel algorithms and architectures*, SPAA '03, pages 230–239, New York, NY, USA, 2003. ACM.

[121] M. Sichitiu. Mobility Models for Ad Hoc Networks. In *Guide to Wireless Ad Hoc Networks*, Computer Communications and Networks, pages 1–18. Springer London, 2009. 10.1007/978-1-84800-328-6_10.

[122] NCTUns 6.0 Network Simulator and Emulator [online]. Available: `http://nsl.csie.nctu.edu.tw/nctuns.html`, (August 2012).

[123] I. Stepanov, J. Hähner, C. Becker, J. Tian, and K. Rothermel. A Meta-Model and Framework for User Mobility in Mobile Networks. In *Procedeengs of the 11th International Conference on Networking 2003 (ICON 2003)*, pages 231–238, 2003.

[124] I. Stepanov, P. Marron, and K. Rothermel. Mobility modeling of outdoor scenarios for manets. In *Proceedings of the 38th Annual Simulation Symposium (ANSS 38)*, pages 312–322, 2005.

[125] S.H. Strogatz. Exploring complex networks. *Nature*, 410(6825):268–276, 2001.

[126] M. Sánchez and P. Manzoni. ANEJOS: a Java based simulator for ad hoc networks. *Future Generation Computer Systems*, 17(5):573 – 583, 2001.

[127] J. Tian, J. Hahner, C. Becker, I. Stepanov, and K. Rothermel. Graph-based mobility model for mobile ad hoc network simulation. In *Proceedengs 35th Annual Simulation Symposium*, pages 337–344, 2002.

[128] V. Tolety. Load Reduction in Ad Hoc Networks Using Mobile Servers. Master's thesis, Colorado School of Mines, Colorado, 1999.

[129] Ns2 trace reader for Ns3 simulator [online]. Available: `http://www.nsnam.org/docs/release/3.15/doxygen/classns3_1_1_ns2_mobility_helper.html`, (August 2012).

[130] 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, March 2005.

[131] D. J. Watts and S. H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393(6684):440–442, Jun 1998.

[132] O. Wibling, J. Parrow, and A. Pears. Ad Hoc Routing Protocol Verification Through Broadcast Abstraction. In F. Wang, editor, *Formal Techniques for Networked and Distributed Systems - FORTE 2005*, volume 3731 of *Lecture Notes in Computer Science*, pages 128–142. Springer Berlin/Heidelberg, 2005.

[133] DEMOS (Domestic Environment Monitoring with Opportunistic Sensor networks) [online]. Available: `http://www.laccir.org/laccir/r1208lac001/Project/tabid/497/Default.aspx`, (August 2012).

[134] S. Yang, X. Yang, C. Zhang, and E. Spyrou. Using social network theory for modeling human mobility. *IEEE Network*, 24(5):6–13, September 2010.

[135] J. Yeo, D. Kotz, and T. Henderson. CRAWDAD: a community resource for archiving wireless data at Dartmouth. *SIGCOMM Computer Communication Review*, 36:21–22, April 2006.

[136] J. Yoon, M. Liu, and B. Noble. Random waypoint considered harmful. In *IEEE INFOCOM 2003. Twenty-second Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE Cat. No.03CH37428)*, volume 2, pages 1312–1321. IEEE, 2003.

[137] J. Yoon, B. D. Noble, M. Liu, and M. Kim. Building realistic mobility models from coarse-grained traces. In *Proceedings of the 4th international conference on Mobile systems, applications and services*, MobiSys '06, pages 177–190, New York, NY, USA, 2006. ACM.

[138] Z. R. Zaidi, B. L. Mark, and R. K. Thomas. A two-tier representation of node mobility in ad hoc networks. In *Proceedings First Annual IEEE Communications Society Conference Sensor and Ad Hoc Communications and Networks IEEE SECON 2004*, pages 153–161, 2004.

[139] X. Zeng, R. Bagrodia, and M. Gerla. GloMoSim: a library for parallel simulation of large-scale wireless networks. In *Proceedings of the twelfth workshop on Parallel and distributed simulation*, PADS '98, pages 154–161, Washington, DC, USA, 1998. IEEE Computer Society.

[140] X. Zeng, R. Bagrodia, and M. Gerla. GloMoSim: a library for parallel simulation of large-scale wireless networks. *SIGSIM Simul. Dig.*, 28:154–161, July 1998.

[141] W-X. Zhou, D. Sornette, R. A. Hill, and R. I M Dunbar. Discrete hierarchical organization of social group sizes. *Proc Biol Sci*, 272(1561):439–444, Febrary 2005.