

Behavioral Mobility Model with Geographic Constraints

Razvan Beuran
Hokuriku StarBED Technology Center
National Institute of Information
and Communications Technology
Nomi, Ishikawa, Japan
razvan@nict.go.jp

Shinsuke Miwa
Hokuriku StarBED Technology Center
National Institute of Information
and Communications Technology
Nomi, Ishikawa, Japan
danna@nict.go.jp

Yoichi Shinoda
Research Center For Advanced
Computing Infrastructure
Japan Advanced Institute of
Science and Technology
Nomi, Ishikawa, Japan
shinoda@jaist.ac.jp

Abstract—Wireless networks offer the opportunity for the users to move as they use the network in scenarios such as voice and video communication, search-and-rescue operations, etc. User mobility typically takes place in inhabited areas, hence it is subject to constraints regarding the road and building infrastructure in those areas. In this paper we propose a mobility model that takes into account such realistic topographical data. The model is driven by rules that replicate the behavior of mobile users as they proceed towards a destination. We implemented this mobility model in the QOMET wireless network emulation set of tools, and we evaluated it from several perspectives, including through a 50-node OLSR case study in an urban environment. The results demonstrate the advantages of our mobility model in terms of realism and practical applicability.

Keywords—mobility models; geographic constraints; wireless networks; mobile networks; network emulation.

I. INTRODUCTION

Wireless network technologies make possible the mobility of the wireless nodes as they continuously use the network. This capability exists for legacy technologies such as IEEE 802.11a/b/g/n (WLAN), or IEEE 802.15.4 (ZigBee), and also for newer technologies such as IEEE 802.16e (Mobile WiMAX). Mobile ad hoc networks and mesh networks, sensor networks, and the generic class of delay-tolerant networks are typical examples of applications of mobile networking in real-life circumstances.

In such real-world scenarios node mobility is not free, as it is subject to the topographical constraints of the area where motion takes place, which for inhabited areas comes down to the structure of roads and buildings.

Network researchers have a lot of mobility models to chose from for network experiments, as the surveys in [8] and [2] show. However, due to the geographic constraints mentioned above, many of the simple mobility models such as random waypoint do not realistically reflect the movement of nodes in real circumstances.

The need to employ realistic mobility models is emphasized by the impact the choice of mobility models has on wireless network performance, as it has been demonstrated in studies such as [3] and [14]. This impact is particularly significant for urban and indoor environments with difficult

propagation conditions, or for short-range wireless technologies, when small-scale movement has a stronger effect on communication.

The work to improve mobility model realism at small scale has started initially by modifications of the classical random waypoint model, as it was done in [4]. Several other approaches followed, including the behavioral mobility model proposed in [12], that we denote by BMM. Motion is described in BMM by using rules that “represent expectations of how mobile individuals react and are influenced by their environment”.

In this paper we propose a mobility model called *behavioral mobility model with geographic constraints* (denoted by BMM-GC) that extends and improves the work in [12]. The extensions refer mainly to the integration of geographic constraints into BMM, which combined with the scalability of the model make it possible to use BMM-GC to generate realistic mobility patterns for road and building topographies.

The contributions of this paper are as follows:

- We provide the details of the BMM-GC mobility model that we propose;
- We discuss the integration of BMM-GC with a wireless network emulation testbed;
- We evaluate the model qualitatively and quantitatively, including through a 50-node OLSR case study that represents a typical scenario for BMM-GC.

The paper is organized as follows. In Section II we introduce the framework and principles of the BMM-GC mobility model that we propose. In Section III we discuss various practical aspects related to the implementation of BMM-GC and its integration with a wireless network emulation platform. We then evaluate the model by presenting example trajectories, a comparison with other approaches, and an OLSR case study in Section IV. The paper ends with a section of conclusions and references.

II. MOBILITY MODEL

The original behavioral motion model presented in [12] is based on the idea that the movement of an individual results from the composition of atomic behaviors, such

as following a path, avoiding obstacles, mutual collision avoidance, etc. Each of these atomic behaviors can be described by *behavioral rules* that are expressed as attractive or repulsive forces. Thus, a mobile entity will be attracted by its destination, and it will be rejected by walls and obstacles. The composition of these forces determines the actual node movement.

The main advantages of behavioral mobility compared to classical mobility modeling are as follows:

- *Abstraction*: By using rules, one creates an abstract view on mobility, in which node behavior can be modified simply by changing either the set of rules that govern its mobility, or only the values of their parameters;
- *Dynamism*: Since the mobile entities are driven by behavioral rules, they continuously interact with their changing environment (obstacles and other entities) in a dynamic way, without any external intervention.

Next we briefly introduce the main features of the original mobility model framework (BMM). Then we detail our extended mobility model (BMM-GC). This is followed by a discussion of several particular issues related to BMM-GC and how it compares to BMM.

A. BMM framework

BMM can be seen more as a theoretical framework for mobility modeling rather than a practical mobility model *per se*. This is because [12] focuses mainly on providing the general design of behavioral rule-based mobility, and touches only lightly on implementation details.

The authors of BMM do present two practical case studies, one for individual pedestrian mobility, and one for group mobility. Since we have so far only focused on individual mobile entities, we shall limit our presentation to the individual pedestrian case, but extensions of our work to group mobility are possible through similar mechanisms as the ones mentioned in the BMM paper.

The following rules are applied when the BMM framework is used for individual pedestrians:

- 1) *Path following*: Rule that results in an attraction force towards the destination;
- 2) *Wall avoidance*: Rule that results in a repulsive force normal to walls;
- 3) *Obstacle avoidance*: Rule that results in an avoidance action around obstacles by following those edges that minimize the detour (obstacles are defined in [12] as disks or convex polygons);
- 4) *Mutual avoidance*: Rule that results in nodes avoiding collision with each other.

The behavior of an individual pedestrian that follows the above rules can be summarized as follows. The individual will follow a path towards its destination, while avoiding walls and obstacles. The individual will also avoid collision

with other individuals it may encounter. In this context an individual is modeled as a disk with a given radius.

Each of the above rules will produce a force and its corresponding acceleration, and these accelerations are combined into a resulting one that determines the motion. However, if the acceleration value exceeds a certain threshold, only the high-priority component will be used, with priority being defined as highest for wall avoidance, followed by mutual avoidance and path following; the priority for obstacle avoidance is not specified in [12], but we consider it should be in-between mutual avoidance and path following.

B. Our mobility model (BMM-GC)

The feasibility of BMM has been demonstrated in [12], but only for very basic scenarios, such as movement on a straight corridor with one disk-shaped obstacle. While such simple small-scale environments may be representative for certain types of indoor sensor networks, they are certainly not characteristic for the more generic types of wireless networks the research community focuses on.

In particular, we targeted network experiments in small-to-medium size urban areas with sizes of 0.1 to 1 km on each axis (hence, roughly with an area between 0.01 and 1 km²) when using medium-range wireless network technologies such as IEEE 802.11a/b/g. The BMM model mentioned in Section II-A cannot be used directly for this kind of scenarios, first of all because of their increased topographical complexity. The limitations of BMM motivated our work to improve and extend this model, so that the behavioral mobility paradigm can be applied in practice for a wider range of network experiments.

Our main extension of BMM refers to operation in realistic environments with geographic constraints, hence we named our model BMM-GC. Our model too uses behavior rules to describe user mobility, but modifies them accordingly so as to take into account the geographic constraints of realistic environments. Since BMM-GC is based on the BMM framework, our model is similar to BMM in broad lines, and we will point out the differences between them as we describe BMM-GC in what follows.

Our mobility model has the following components, as shown in Figure 1:

- Destination attraction (F_{DA});
- Object rejection (F_{OR}) and object avoidance (F_{OA});
- Mutual avoidance (F_{MA}).

Note that, differently from BMM, we avoid the distinction between walls and obstacles, and use the term “object” to designate all the geographic elements that will restrict motion, such as buildings, or building areas that define the shape of the roads. This more general notation can still be applied to indoor environments, with walls being a particular type of object.

Next we introduce each of the component forces of the BMM-GC model when applied to a given mobile node.

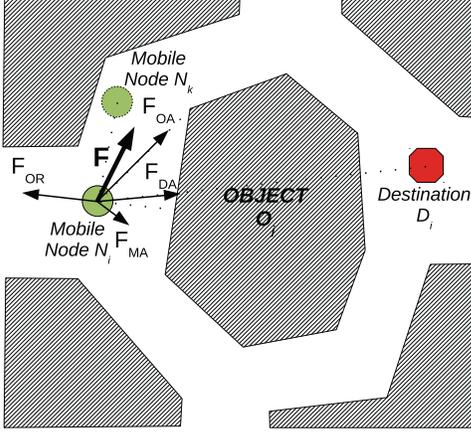


Figure 1. Behavioral mobility forces in the BMM-GC model.

1) *Destination attraction*: This force is the main force driving the motion of the node. F_{DA} is a force oriented from the current position of the node towards its destination. The acceleration \vec{a}_{DA} corresponding to this force at time t for the node N_i is given by:

$$\vec{a}_{DA}(t) = \alpha[v_{i0} \frac{\vec{D}_i - \vec{N}_i(t)}{\|\vec{D}_i - \vec{N}_i(t)\|} - \vec{v}_i(t)], \quad (1)$$

where α is a calibration parameter, v_{i0} is the target speed of the node, \vec{D}_i is the destination of the node, $\vec{N}_i(t)$ is the position of the node at time t , and $\vec{v}_i(t)$ is the node's instantaneous velocity. F_{DA} is essentially equivalent with the path following force in BMM.

2) *Object rejection*: This force prevents nodes from entering into objects. F_{OR} is perpendicular on the object edge that is closest to the node, and is oriented from the object towards the node. The acceleration \vec{a}_{OR} corresponding to this force at time t for the node N_i with respect to an object O_j is given by the following equation:

$$\vec{a}_{OR}(t) = \begin{cases} \beta[\vec{N}_i(t) - \vec{P}_{nearest}(O_j)], & \text{if } d_{ij} < d_{R0} \\ 0, & \text{otherwise} \end{cases}, \quad (2)$$

where β is a calibration parameter, $\vec{P}_{nearest}(O_j)$ is the point on the contour of the object that is nearest to the node, and d_{ij} is the distance between the node and the object. The acceleration is null if d_{ij} exceeds a pre-defined rejection threshold d_{R0} , in which case the object has no influence on the node. F_{OR} is a generalized version of the wall avoidance force in BMM.

3) *Object avoidance*: This force ensures that a node heading towards a destination will move on the shortest path around an object that is in its way. F_{OA} follows the direction that connects the current position of the node with the object vertex for which the path to destination along the contour of the object is shortest. The acceleration \vec{a}_{OA} corresponding

to this force at time t for the node N_i with respect to an object O_j is given by:

$$\vec{a}_{OA}(t) = \begin{cases} \gamma[\vec{V}_{min}(O_j) - \vec{N}_i(t)], & \text{if } d_{ij} < d_{A0} \\ 0, & \text{otherwise} \end{cases}, \quad (3)$$

where γ is a calibration parameter, and $\vec{V}_{min}(O_j)$ is the object vertex that minimizes the length of the path to destination. As in the case of rejection, object avoidance doesn't take place if the distance between the node and the object exceeds a pre-defined avoidance threshold d_{A0} . F_{OA} is a generalized version of the obstacle avoidance force in BMM, but we use object vertexes instead of edges when computing the avoidance trajectory. Thus, we minimize the traveled path by determining the object vertex for which the following sum is minimum: the distance between the node and the vertex, plus the lengths of subsequent object edges that form the contour when heading towards the destination, plus the distance between the first object vertex from which the destination can be reached and the destination.

4) *Mutual avoidance*: This force ensures that two nodes moving in the vicinity of each other do not collide, which is an important feature in order to create movement that is realistic even at small scale. F_{MA} will push laterally two nodes that approach each other so that they avoid the collision. The acceleration \vec{a}_{MA} corresponding to this force at time t for the node N_i with respect to the node N_k is given by:

$$\vec{a}_{MA} = \begin{cases} \delta \frac{\theta_{ik}}{d_{ik}} \vec{u}_{i\perp}(t), & \text{if } d_{ik} < d_{M0} \\ 0, & \text{otherwise} \end{cases}, \quad (4)$$

where δ is a calibration parameter, $\vec{u}_{i\perp}(t)$ is a unit vector perpendicular on the instantaneous velocity $\vec{v}_i(t)$ of the node N_i , θ_{ik} is the angle in the interval $[0, \pi]$ formed by the respective velocities of the two nodes, and d_{ik} is the distance between the two nodes. Mutual avoidance is null if the distance between the nodes exceeds a pre-defined mutual avoidance threshold d_{M0} . The orientation of $\vec{u}_{i\perp}(t)$ is chosen so as to minimize deviation. The parameter θ_{ik} ensures that pedestrians react more strongly to the pedestrians coming from opposite direction then to the ones going in a similar direction. F_{MA} is essentially equivalent with the mutual avoidance force in BMM.

Once the acceleration values are computed as given by Equations (1) to (4), they are merged using vector composition rules to obtain the resulting acceleration $\vec{a}(t)$. The kinetic equations are then applied to update the velocity and the position of the node at the next moment of time, $\vec{v}_i(t + \Delta t)$ and $N_i(t + \Delta t)$, respectively:

$$\vec{v}_i(t + \Delta t) = \vec{v}_i(t) + \vec{a}(t)\Delta t, \quad (5)$$

$$\vec{N}_i(t + \Delta t) = \vec{N}_i(t) + \vec{v}_i(t)\Delta t + \vec{a}(t)(\Delta t)^2/2. \quad (6)$$

C. Discussion

The description of the behavioral rules in BMM-GC given in Section II-B emphasizes the fact that rules 1) and 4) are essentially equivalent to their counterparts in BMM, whereas rules 2) and 3) have been generalized so that they can be applied to complex environments with geographic constraints. We discuss below several issues related to the BMM-GC model and how it compares to BMM.

In BMM-GC we only considered polygonal objects, because this is how topographical information is represented, but the disks that are discussed in BMM can be approximated by polygons and consequently dealt with in the same way as the polygons in BMM-GC. Topographical shapes can be either convex or concave polygons. Although BMM-GC does not require the polygons to be convex, whereas BMM does, the presence of concave polygons can change the outcomes that we have discussed so far, such as producing a non-optimal path when using Equation (3). In this case a concave polygon can be replaced – for computational purposes – by the minimum convex polygon that includes the corresponding concave polygon. One other issue with Equation (3) is that it does only a local trajectory optimization, since not all objects are taken into account; we consider this to be an acceptable trade-off given the computation resources that would be required otherwise.

An issue shared by both BMM-GC and BMM is that, while in theory node movement is continuous, in practice a discrete time step Δt is used in Equations (5) and (6). If the time step is too large, the node trajectory will not be smooth. Moreover, the node may be unable to enter narrow roads, especially when moving fast. Therefore, in practice, the time step which is most appropriate for a scenario must be determined depending on the topography of the area involved. In our experience, a time step of 500 ms is sufficient for most pedestrian scenarios, but in some cases 250 ms or even 125 ms steps were required, which implies a longer computation time. A possibility would be to chose this time step adaptively and independently for each mobile node, depending on the conditions it encounters.

Another issue that is common to both BMM and BMM-GC is that the user has no control over the path a mobile node will take to reach a certain destination. The solution proposed in BMM to achieve this control if desired, and that we follow, is to introduce a series of “waypoints” that become intermediate destinations as the mobile node proceeds towards its final destination. This solution can also be used to work around potential trajectory issues in complicated scenarios.

Some limitations of BMM, and how we addressed them in BMM-GC follow. Although BMM proposes the use of bounds on the resulting acceleration obtained by composing the atomic accelerations described in the model, we determined that this is not a sufficient restriction in complex

environments, since moving entities that are too close to an object may enter the object despite this acceleration bound. Therefore we found it necessary to test whether the future position of a node, as computed by Equation (6), is within an object or not, and limit the node movement accordingly if this is the case.

One other problem that we noticed with BMM is that the forces that act upon a mobile entity may sometimes reach a quasi-equilibrium, which causes movement to become very slow or even stop, even though the node has not yet reached its destination. We propose to detect such a state of relative lack of motion, and apply a small random displacement to break the temporary equilibrium state. Of course this action has to be done in such a way that the node does not enter neighboring objects. It is worth mentioning the fact that the use of discrete time discussed above actually limits this effect; the larger the time step the smaller the probability to reach an equilibrium state is.

III. MODEL IMPLEMENTATION

We integrated the BMM-GC model with the wireless network emulation testbed called QOMB, so that it can be used directly for experiments in a straightforward manner. The resulting mobility traces can also be exported and used in other tools, such as the network simulator Ns-2 (see Section III-A2). In what follows we shall first describe QOMB, followed by an overview of our implementation.

A. QOMB

The general architecture of the wireless network emulation testbed QOMB, and utilization examples in the context of WLAN were presented in [6]. QOMB is created by integrating a large-scale wired-network experiment infrastructure called StarBED with the wireless network emulation set of tools called QOMET.

1) *StarBED*: A network testbed located at the Hokuriku StarBED Technology Center of the National Institute of Information and Communications Technology, in Ishikawa, Japan [13]. StarBED makes more than 1100 interconnected PCs available for experiments, and it represents the infrastructure of QOMB.

The main experiment-support software tool for StarBED is called SpringOS, which allows users to easily perform complex experiments with a large number of hosts [13]. The main two functions of SpringOS are:

- *Experiment preparation*: Configuring the experiment hosts and network so that they are ready for experiment execution;
- *Experiment execution*: Carrying out the effective experiment by executing in the required order the necessary commands on the experiment hosts.

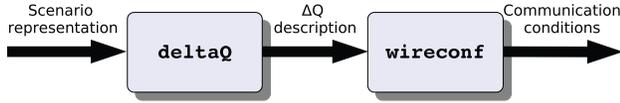


Figure 2. Relationship of the QOMET libraries `deltaQ` and `wireconf`.

2) *QOMET*: A set of tools for wireless network emulation in complex scenarios including mobility (the acronym stands for “Quality Observation and Mobility Experiment Tools”). QOMET provides the necessary mechanisms for performing wireless network emulation in a distributed manner by reproducing the communication conditions between the emulated wireless nodes in the experiment. QOMET relies on the experiment management mechanisms of StarBED for its distributed execution.

The most important components of QOMET in the context of this paper are the libraries called `deltaQ` and `wireconf`. The relationship between these two libraries is shown in Figure 2, where the notation “ ΔQ description” is used to represent the network quality degradation parameters that characterize the network at each moment of time.

The `deltaQ` library is in charge of computing the communication conditions between wireless nodes given a user-defined scenario. The scenario representation specifies characteristics such as:

- Properties of the wireless nodes: position, network technology parameters, mobility patterns, and so on;
- The environment in which the wireless nodes are placed: attenuation, shadowing, road and building topography, etc.

The network degradation computed by `deltaQ` is recreated during the real-time emulation experiment as the communication conditions between the wireless nodes emulated on StarBED; this is achieved by using the `wireconf` library. This library is in charge of controlling network degradation characteristics (bandwidth limitations, packet loss, delay and jitter) so that the communication between nodes in the wired network of StarBED takes place similarly to that in the wireless network in the user-defined scenario. The `wireconf` library uses the `deltaQ` library in real time to compute dynamically the communication conditions under the effect of contention.

Mobility is typically pre-computed by `deltaQ` before an emulation experiment, so as not to waste the real-time computation resources needed during the effective experiment (if needed, mobility can also be computed dynamically as the experiment takes place). One important function of `deltaQ` is the possibility to export the generated mobility data, in formats such as that supported by Ns-2; the export settings are specified via command-line parameters. This makes it possible to use `deltaQ` standalone from the full emulation framework of QOMB, as a mobility generator that provides input to other tools.

- 1: let N_i be the current node
- 2: compute \vec{a}_{DA} for N_i
- 3: **for all** objects O_j in the neighborhood of node N_i **do**
- 4: compute \vec{a}_{OR} with respect to O_j
- 5: compute \vec{a}_{OA} with respect to O_j
- 6: **end for**
- 7: **for all** nodes N_k in the neighborhood of node N_i **do**
- 8: compute \vec{a}_{MR} regarding N_k
- 9: **end for**
- 10: compute the resulting acceleration \vec{a} at time t
- 11: compute new velocity and position of N_i at time $t + \Delta t$

Figure 3. Pseudo-code of the BMM-GC implementation.

B. Implementation overview

As mentioned in Section III-A2, QOMET already supports several mobility models (such as linear motion and random walk). This functionality is implemented through a mobility computation engine that is part of the `deltaQ` library. This module computes the new position of a node after the time interval Δt based on its position at time t by using the corresponding motion equations or algorithms, which are applied iteratively to the mobile nodes.

The BMM-GC model was implemented in the `deltaQ` library of QOMET by leveraging the presence of the mentioned mobility computation engine. Practically this required to implement the functionality that corresponds to the four motion rules presented in Section II-B. The pseudo-code that describes the implementation is given in Figure 3.

The implementation is relatively straight-forward given our discussion in Section II-B. One thing to note is that we compute object rejection and avoidance in the same loop because they typically refer to the same objects.

Although for simplicity reasons we did not mention this explicitly in Figure 3, all the considerations regarding limiting acceleration values and preventing nodes from entering objects must be taken into account at steps 10 and 11 in the pseudo-code.

C. GIS support

One required feature for the implementation of BMM-GC was the support for realistic road and building environments. We achieved this by implementing support in `deltaQ` for importing map data in the JPGIS format, which is a Japanese version of the GIS (Geographic Information System) format. JPGIS data is freely available for the entire Japan at a scale of 1/2500 from the Geospatial Information Authority of Japan [9].

The JPGIS data is used to create within QOMET a “virtual world” in which the wireless nodes move and communicate with each other. This topographical information is integrated with `deltaQ` in such a way that the BMM-GC implementation can use this information to create the geographical constraints that limit the motion of the nodes.

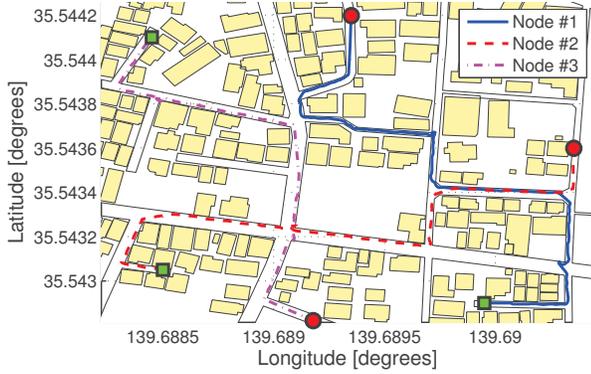


Figure 4. Trajectories generated by BMM-GC for a 3 pedestrian scenario.

IV. MODEL EVALUATION

In this section we proceed to evaluate BMM-GC from various perspectives in order to demonstrate its practicality.

A. Trajectory examples

In Figure 4 we show examples of trajectories generated by BMM-GC for 3 pedestrians. The urban area used has a size of about 200 x 170 m and uses map information for Kawasaki, Japan. The starting positions are marked by a green square, and the destinations are marked by a red circle, respectively. These example trajectories illustrate how the nodes move from residential zones towards the main streets of the city area. For example, node #1 starts from a zone in the lower right-hand side corner of the figure and follows a realistic path toward its destination at the top of the figure. Nodes #2 and #3 have similarly realistic trajectories towards their respective destinations.

The parameters that we used in BMM-GC for pedestrian motion are shown in Table I. We used 1 as the default value for the calibration parameters α , β , γ , and δ . We also used the typical value 1 m/s for the pedestrian speed [12].

The object rejection and avoidance thresholds were set to 0.75 m and 1.5 m, so that the avoidance action starts first, and the rejection action only occurs if a node gets too close to an object. While larger values would lead to smoother movement around object corners, we determined through experiments that smaller values, such as the ones we used, are required in order to enable the nodes to “maneuver” more easily on narrow roads. The mutual avoidance threshold was set to 2 m, which is the double of the radius of the individual “territory”, which we consider to be 1 m. The computation step was set to 0.5 s, which we determined to be optimal for most scenarios we studied.

Table I
PEDESTRIAN MOTION PARAMETERS FOR BMM-GC

Parameter name	Value
Calibration parameters ($\alpha, \beta, \gamma, \delta$)	1
Target node speed (v_{i0})	1 m/s
Object rejection threshold (d_{R0})	0.75 m
Object avoidance threshold (d_{A0})	1.5 m
Mutual rejection threshold (d_{M0})	2 m
Computation step (Δt)	0.5 s

B. Comparison

Next we present qualitative and quantitative comparisons of BMM-GC with other mobility approaches that target environments with geographic constraints.

One of the classical solutions for modeling node mobility in areas with buildings is to use a graph representation of the area, such as the Voronoi graphs, as proposed in [10]. Dijkstra’s algorithm is then used to find the shortest path in the graph between a starting position and a destination. Such a model does not provide realistic results at small scale, since nodes can only use graph edges to move, and it also doesn’t take into account mutual collisions between nodes. A related model is proposed in [1], which increases the graph complexity and potential accuracy by adding so-called “anchors”, for instance near entrances and building corners, so as to augment the realism of the generated trajectory. Nevertheless, this increases computation complexity, and mutual effects between nodes are still not taken into account.

The model proposed in [11] uses a combined diffusion and steepest gradient algorithm to generate node trajectories. This approach requires to create a layout map matrix that defines which areas are accessible to nodes and which are not. Therefore this model cannot be directly applied to city map information. Moreover, the diffusion algorithm requires to process the entire movement area, which is unfeasible for large scenarios. By contrast, BMM-GC only requires local knowledge to generate the trajectory, and areas in which nodes do not enter will not be processed.

Computational complexity is an important characteristic of any motion generation algorithm, especially when large scenarios are targeted, or when there are real-time generation constraints. Let us analyze first the case of BMM-GC. We assume the total number of nodes is N , the total number of object vertexes is W and the total number of computation steps is S . The worst-case complexity of BMM-GC is:

$$O((N + 2NW + N^2) \times S). \quad (7)$$

The N component comes from the destination attraction computation, the $2NW$ component comes from the object rejection and avoidance computation, while the N^2 component comes from mutual rejection computation, all done for each node in the scenario. These computations are repeated at each step, hence the presence of the multiplier S (for reference, considering a computation step of 0.5 s and a

scenario duration of the order of 10^2 s, then S is of the order of 10^3 .

Equation (7) represents a worst case because, in practice, at a certain moment a node will only interact with a small number of objects and nodes which are in its vicinity. For such a case, complexity becomes dominated by $N \times S$, as follows:

$$O((1 + 2N + N) \times S) \cong O(N \times S). \quad (8)$$

For comparison, according to [1] (if we assume each node has only one destination during the simulation period), the complexity of the Voronoi graph approach, which is the most representative alternative approach, is given by:

$$O(W \log W + N \times W^2) \quad (9)$$

Since the complexity of the Voronoi graph approach is dominated by $N \times W^2$, we conclude that our approach has better performance in large-scenarios with a large number of objects (hence with a high value of W), which is exactly the kind of scenarios we target. On the other hand, the Voronoi graph approach has a better overall performance in simple scenarios with a small number of objects.

C. OLSR Case Study

To illustrate the potential of the BMM-GC model we summarize here an experiment that assessed OLSR protocol performance in a MANET scenario with 50 mobile wireless nodes [7]. All the nodes but one (which is considered to be the gateway) moved to predefined locations by using the BMM-GC model in an urban topography. Our extensions of BMM as presented in Section II-B were effective in generating realistic motion trajectories in such settings.

The building areas that represent the geographic constraints of the environment and define the roads were imported from JPGIS map data for an area in Kawasaki, Japan with a size of about 400 x 300 m. A snapshot half-way through the 500 s experiment is shown in Figure 5. The starting position of the nodes is marked with a green square, and their destinations with red circles. The figure also depicts the links between nodes as discovered by the OLSR protocol executed on QOMB.

In this particular experiment we modified the value of the transmission power of the emulated nodes from very low values (1 mW) to the largest value allowed by the IEEE 802.11b standard (100 mW). The evaluation of the resulting OLSR network topology in terms of average route length, maximum route length and number of disconnected nodes made it possible to identify that the transmit power of 10 mW provided optimum performance for this scenario.

Although this case study is but an example, similar topographies can be used to make realistic experiments in urban areas, for instance in order to estimate how a certain network application or protocol will behave in a real target zone (e.g., in connection with disaster management or

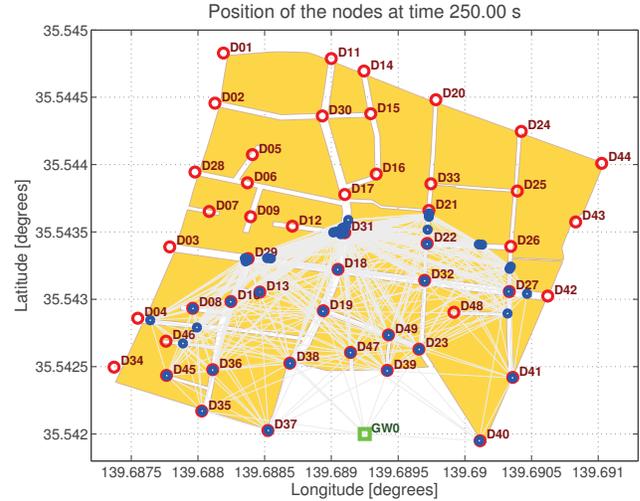


Figure 5. OLSR case study using the BMM-GC mobility model in an urban topography.

recovery activities), simply by importing the corresponding map information in QOMET.

Note that we do not investigate in this paper the variation of network conditions depending on a particular mobility model because this issue was already investigated in [12] in connection with BMM, on which our model is based. Moreover, a thorough and more general analysis of this aspect is available in [3] and [14].

V. CONCLUSION

In this paper we have presented a behavioral motion model with geographic constraints (BMM-GC) that computes trajectories of mobile nodes based on predefined destinations in a realistic area with roads and buildings structures. The computation uses behavioral rules such as attraction to destination, object rejection and avoidance and mutual avoidance, while taking into account the geographic constraints of the virtual environment in which the mobile nodes are located. The algorithm has good computational complexity compared to other methods, especially for large-scale scenarios.

Using an OLSR case study we showed that by using this model various realistic emulation experiments can be performed on the wireless network emulation testbed called QOMB, which integrates the BMM-GC model by means of the software tool called QOMET. Through the export function of QOMET, the output of the BMM-GC model can also be used in other experiment tools, such as the Ns-2 network simulator.

REFERENCES

- [1] S. Ahmed, G. C. Karmakar, J. Kamruzzaman, *An Environment-Aware Mobility Model for Wireless Ad Hoc Network*, Journal of Computer and Telecommunications

- Networking, Elsevier, Vol. 54, Issue 9, June 2010, pp. 1470-1489
- [2] F. Bai, A. Helmy, *A Survey of Mobility Models in Wireless Adhoc Networks*, in "Wireless Ad Hoc and Sensor Networks", Kluwer Academic Publishers, 2004.
- [3] F. Bai, N. Sadagopan, A. Helmy, *The IMPORTANT Framework for Analyzing the Impact of Mobility on Performance of Routing for Ad Hoc Networks*, Ad Hoc Networks Journal, Elsevier, Vol. 1, No. 4, pp. 383-403, November 2003.
- [4] C. Bettstetter, *Smooth is better than sharp: A random mobility model for simulation of wireless networks*, 4th ACM International Workshop on Modeling, Analysis, and Simulation of Wireless and Mobile Systems (MSWiM01), Rome, Italy, 15-20 July 2001, pp. 19-27.
- [5] R. Beuran, J. Nakata, T. Okada, L. T. Nguyen, Y. Tan, Y. Shinoda, *A Multi-purpose Wireless Network Emulator: QOMET*, 22nd IEEE Intl. Conf. on Advanced Information Networking and Applications (AINA 2008), FINA 2008 symposium, Okinawa, Japan, March 25-28, 2008, pp. 223-228.
- [6] R. Beuran, L. T. Nguyen, T. Miyachi, J. Nakata, K. Chinen, Y. Tan, Y. Shinoda, *QOMB: A Wireless Network Emulation Testbed*, IEEE Global Communications Conference (GLOBECOM 2009), Honolulu, Hawaii, USA, November 30-December 4, 2009.
- [7] R. Beuran, L. T. Nguyen, Y. Shinoda, *QOMB Wireless Network Emulation Testbed: Evaluation and Case Study*, 5th ACM International Workshop on Wireless Network Testbeds, Experimental Evaluation and Characterization (WiNTECH 2010), in conjunction with MobiCom 2010, Chicago, Illinois, USA, September 20-24, 2010, pp. 1-8.
- [8] T. Camp, J. Boleng, V. Davies, *A Survey of Mobility Models for Ad Hoc Network Research*, Wireless Communications and Mobile Computing, vol. 2, no. 5, August 2002, pp. 483-502.
- [9] Geospatial Information Authority of Japan (GSI), *Inspection and download of basic map information* (in Japanese), <http://www.gsi.go.jp/kiban/etsuran.html>
- [10] A. Jardosh, E. M. Belding-Royer, K. C. Almeroth, S. Suri, *Real world environment models for mobile ad hoc networks*, IEEE Journal on Special Areas in Communications - Special Issue on Wireless Ad hoc Networks 23 (3), 2005, pp 622632.
- [11] J. Kammann, M. Angermann and B. Lami, *A New Mobility Model Based on Maps*, IEEE Semiannual Vehicular Technology Conference (VTC Fall 2003), Orlando, USA, October 6-9, 2003.
- [12] F. Legendre, V. Borrel, M. D. De Amorim, S. Fdida, *Reconsider microscopic mobility modeling for self-organizing networks*, IEEE Communications Magazine, vol. 20, No. 6, Nov.-Dec. 2006, pp 4-12.
- [13] T. Miyachi, K. Chinen, Y. Shinoda, *StarBED and SpringOS: Large-scale General Purpose Network Testbed and Supporting Software*, Intl. Conf. on Performance Evaluation Methodologies and Tools (Valuetools 2006), ACM Press, Pisa, Italy, October 2006.
- [14] P. Prabhakaran, R. Sankar, *Impact of Realistic Mobility Models on Wireless Networks Performance*, IEEE Intl. Conf on Wireless and Mobile Computing, Networking and Communications (WiMob'2006), Montreal, Canada, June 19-21, 2006, pp. 329-334.