Geographic Constraint Mobility Model for Ad Hoc Network

Sabbir Ahmed  
Monash University  
sabbir@infotech.monash.edu.au

Gour C. Karmakar  
Monash University  
gour@infotech.monash.edu.au

Joarder Kamruzzaman  
Monash University  
joarder@infotech.monash.edu.au

Abstract

In this paper, we propose a mobility model and present its simulation tool to generate realistic mobility traces for mobile ad hoc network. The mobility model is capable of creating realistic node movement pattern in the presence of geographic constraints by exploiting the concepts of anchors. The model dynamically places anchors depending upon the context of the environment through which nodes are guided to move towards the destination, and obstacles of arbitrary shapes with or without doorways and any existing pathways, in full or part of the terrain can be incorporated which makes the simulation environment more realistic. The characteristics of the proposed mobility model tested on a real world university campus map at various movement patterns are presented that illustrate the impact of the mobility model on the performance of a routing protocol and usefulness of the proposed scenario generation tool.

1. Introduction

Over the last decade, huge research efforts have been dedicated to the theoretical development and analysis of MANET in relation of capacity estimation, routing, scalability, location service, mobility modeling, performance evaluation and others. Such theoretical study can only be verified through test beds, real-world deployment or appropriate simulation tools that emulate practical scenario. Simulation environment is very attractive for evaluating performance of MANET through various metrics, since it reduces the huge cost and time for real world study through test beds or practical deployment. It helps to generate a wide variety of repeatable scenarios which may not be possible in real world deployment. This repeatable scenario generation aids to analyze different perspective of the network performance, through various performance metrics, and readjustment of protocols under consideration is possible [1]. Due to the capability of rapid development of wide range of scenario within short time and scope for refinements, protocol designers use simulation tools for investigating the performance of protocols, but, for accurate evaluation of MANET models, simulation environment must be based on realistic assumptions, without which simulation becomes ineffectual.

One of the key factors for a simulation environment to reflect and assess real-world scenario accurately is to model mobility of nodes in MANET as realistically as possible. After deployment of nodes, this mobility model is responsible for directing the nodes when and where to move. A variety of mobility models that have been proposed in literature for studying mobile network are described in Section 2. Among them Random Waypoint model, Random Walk model and Random Direction model are widely used where nodes are assumed to be deployed in a rectangular sized unobstructed simulation area. Nodes can move any where in the given simulation playground according to the specification of the respective mobility model.

The real world is rather complex and vastly diverse that make the existing mobility models incapable of representing the real world environment properly. The essential constituents of a natural environment are pathways and obstacles. In reality, we often find many obstacles such as buildings, lakes, hills, playgrounds, vegetations that restrict our movement in different ways. Irrespective of the position of pathways, obstacles and their entry points in a real world setup, existing obstacle based mobility model [1] produces a fixed number of restricted paths and entry points to each side of an obstacle and uses shortest path for node movement. However, there are some obstacles where people can move inside e.g. buildings and others they are not expected to enter e.g. hills. There may be defined pathways or no pathways at all for going to a specific region. In many cases people do not use shortest paths to reach the destination. To address this and hence a major unresolved problem “there can be no single model that is the best for all terrains” posed by [1], it is essential to develop a mobility model where node movement is guided considering the elements of a deployment context such as obstacles and existing pathways and a framework which could accurately represent their natural structures.

In this paper, this is accomplished by exploiting the
flexible positioning of anchors and proposing a novel mobility model – Anchor-based Mobility Model for Real World (AMM).

Our proposed mobility model incorporates the above mentioned real world features. Anchors are assumed to be square shaped geographic positions, could be located anywhere in the deployment area depending on the context such as predefined path, obstacles and their entry points, if exist, and thus their positions in a way can reflect the mobility framework in the deployment area. Thus anchors will contribute to generating a graph that will guide the node movement in an appropriate way. A node can reach any position in the simulation area by selecting suitable anchor as the next step. The size of an anchor defines the freedom and flexibility of the node movement. Since according to this model a node selects a random point in the next anchor to be visited, it can produce a more flexible node movement scenario which is able to capture wide variety of movement pattern. The model also reduces surrounding effect [2]. The main contributions of this paper are as follows:

- The proposed model can generate node movement patterns more realistically than existing models and thereby more suitable to simulate MANET in real world deployment.
- Incorporating the facility to introduce any arbitrary shaped obstacles. Obstacles within an obstacle in a nested fashion can also be supported. Moreover, internal floor plan of a building can be adopted as well.
- Placement of doorways at any position in each side of an obstacle (e.g. building) so that nodes can move inside the obstacle through the doors only.
- Feature to adopt existing pathways in any or full portion of the simulation area.
- Computing node movement based on the dynamically generated anchors.
- Allows visualizing the node movement and generated mobility traces.
- A GUI (Graphical User Interface) based program “MobiGen” that includes all the above features and generates node movement scenario file which can easily be used in network simulator NS.

To illustrate the impact of our mobility model on routing protocol, we used a real world environment map of the Clayton campus, Monash University, Australia with infrastructures in exact positions in simulation setup. Greedy Perimeter Stateless Routing (GPSR) [3] which uses geographic location of destination to forward data is used as routing protocol because of its scalability to large networks and low packet overhead. To compare the performance of GPSR for our mobility model with two models widely used in MANET: Voronoi graph based obstacle model [1] as it is capable of generating scenario in presence of obstacles and Random Direction model, some simulation results are presented which exhibit that our proposed model has significant impact on the performance of geographic routing protocol GPSR; most importantly, it simulates the true movement pattern of a node under a real-world scenario more closely than other mobility models.

2. Related work

A variety of mobility models have been proposed in literature for generating node movement pattern. Random Waypoint model is one of the mostly used models where a node randomly selects one location in the simulation space as destination. It then travels towards the destination with a constant velocity chosen uniformly and randomly within the range of \([0, v_{\text{max}}]\) where \(v_{\text{max}}\) represents the maximum allowable velocity and pauses for \(T_{\text{pause}}\), termed as pause time and the whole process continues to repeat until the simulation ends.

To overcome the density waves [4] observed in Random Waypoint model, Random Direction model [5] was introduced. In this model nodes continue to reach the boundary of the simulation area using a randomly chosen direction and speed. One variation, Boundless Simulation Area model was introduced to alleviate the boundary effect observed in Random Direction Model due to pausing at the boundary. Nodes that reach one side of the simulation area continue traveling and reappear on the opposite side which seems unrealistic.

In Random Walk model the nodes change their speed and direction at each time interval. On the other hand Gauss-Markov Mobility model was designed in a way that the value of speed and direction at the \(n^{th}\) instant is calculated on the basis of the value of speed and direction at the \((n-1)^{th}\) instant.

Group mobility models have been introduced to get a picture of the situations where the nodes or a subset of nodes move as a group. Among them Reference Point Group Mobility model [6] is mostly used. In this scheme the random motion of a group of mobile nodes is defined as well as the random motion of each individual node within the group. Variations of Reference Point Group Mobility model are Nomadic Community mobility model where a group maintains a reference point and the nodes within the group can move around the reference point and Pursue Mobility Model which attempts to represent a group of mobile nodes tracking a particular target node just like a group of police chasing a criminal. Detailed survey on mobility models is presented in [7-9].

As illustrated, each of the above mobility models assumes open and unobstructed areas where the nodes can move freely, but in real world situation the existence of such unobstructed scenario is rare and thereby fail to generate the true node movements for a real world environment. A few mobility models [10] that take geographic restrictions into account are Pathway mobility
model and Obstacle mobility model. Pathway Mobility Model uses a predefined map where the vertices represent the buildings and the edges model the streets and freeways between those buildings. Nodes are allowed to move only on the predefined edges. Therefore, it does not consider the actual layouts of the buildings. Variation of this mobility model is City section mobility model that includes Freeway and Manhattan Mobility model where the mobility model mainly focuses the car movement on the freeways connecting different cities and streets in horizontal or vertical directions.

Johansson et al. [11] developed three mobility scenarios to depict the movement of mobile users in real life which includes Conference, Event Coverage and Disaster Relief scenarios. Jardosh et al. [1] also investigated the impact of obstacles on mobility modeling in detail. In their simulation field, a number of obstacles were placed to model the buildings within a campus environment. They used a Voronoi graph to compute the movement path for the nodes. Nodes are only allowed to move on the edges of the Voronoi diagram and reach the destination which is randomly selected within the set of Voronoi vertices using the shortest path.

Even though the geographic restricted mobility models described above give a flavor of realistic movement of mobile nodes, some essential elements of real world mobility are still lacking. The Voronoi graph based model uses only buildings as obstacles and restricts the node movement only on the edges of the graph, thus lacking the randomness of movement in the area, since people are not always expected to move on pre-defined paths. Also the position of doorways in buildings derived from Voronoi graph may stray from the actual position of doors and all sides of a building may not contain doors and consequently fails to illustrate actual environment.

3. Need for a realistic mobility model

Ad hoc network is attractive for communication in battlefields, disaster areas and inter vehicular communication where it embraces a huge number of different deployment contexts. In disaster areas after earthquake, flood, fire or war there might be many buildings, streets and roads destroyed. Established road infrastructure may be totally or partially damaged. Rescue operators may need to save people stuck under demolished or semi-demolished buildings. In absence or demolition of existing infrastructure, peoples’ movement can not be expected to be limited on pathways only and restricting movement may not serve the purpose. Thus a realistic mobility model must be flexible enough to cater a variety of situations. In a university campus to enter a building one has to use doorways only. Also in battlefields soldiers tend to move on a point of interest basis considering the obstacles. For this reason the

pathway and Voronoi graph based mobility models are not suitable for these types of scenarios. Moreover, in Voronoi graph based mobility model, it is not possible to create an obstacle which does not permit to enter and movement path is generated as the boundary edges of the Voronoi cells considering the corners of obstacles. This renders a number of entry points at specific locations and it is more likely not to match with existing doorways of an infrastructure. All the above issues underpin the pressing need for a new mobility model that can simulate a wide range of terrains and provide a movement pattern which resembles more like what happens in real world, and this remains the major focus of the current study.

4. Anchor-based mobility model

In real world when a person finds an obstacle on his way he tries to move around it – human can see obstacles from a distance, they do not necessarily collide on a side and then move along the walls like blind movement [12, 13]. This motivates us to exploit the concept of anchors to illustrate the movement paths by considering the obstacles and existing pathways in mobility model which consequently will lead to node movement more realistically. Once the anchors are generated a node can move to randomly selected destination using shortest path. Several procedures [14, 15] have been proposed in computational geometry to find the shortest path around polygons. Our algorithm follows a similar strategy but instead of using a single point at a corner, it uses anchors, thus reducing the effect of nodes’ higher availability at the borders and corners of obstacles which is termed as surrounding effect [2] as shown in Fig. 1.

Figure 1. (a) Surrounding effect without anchors when a node moves from $P_s$ to $P_d$ (b) Anchors reduces surrounding effect.

It should be noted that our main objective is to develop a realistic mobility model that can adopt doorways in buildings and support pathways rather than finding the shortest path to the destination. Fig. 2 and Fig. 3 show different types of anchors generated in presence of obstacles with or without pathways and how they aid guidance to a node to move from start to destination points, $P_s$ to $P_d$. 
Figure 2. Node movement guided from $P_s$ to $P_d$ by anchors in presence of obstacles with entry point when there is no existing pathway.

Anchors are specific geographic positions to guide the node movement in order to consider the existing pathways and the obstacles including buildings with entry or exit points through doorways. The concept of anchors has been used in Terminode routing protocol [16] where the anchors are generated by selecting a point around which the node density is higher, with the help of monitoring node density throughout the network, in order to guide the packets to destination. However, in our model the computational procedure and usage of anchors is novel as anchors are generated according to the shape of obstacles, placement of doorways and pathways. Depending upon the context of scenario different types of anchors, namely, general anchors, doorway and pathway anchors are generated as shown in Fig. 2 and Fig. 3. A general anchor is created at the side of every convex corner viewed from both inside and outside of an obstacle. As explained in Fig. 2, $c$ is the center of a square shaped external general anchor with width $w$, $d$ distance away from its nearest corner of the obstacle and its diagonal resides on the perpendicular of the tangent at that corner. When multiple obstacles become close enough, anchors are shifted towards its corner and it can be shifted at most $d = w/\sqrt{2}$. If the anchor still intersects with obstacle, the size is reduced further so that the intersection diminishes, until $w > 0$. If $w = 0$ as a result of two objects touching at the corner, the anchor is removed.

By selecting suitable values for $w$ and $d$ it is possible to reduce the surrounding effect, and nodes will be more spatially distributed along the space in between obstacles. Moreover, the framework of using anchors is so flexible and generalized that it can be used to support any existing pathways and doorways for entering and exiting from an obstacle including buildings. It is also possible to support nested obstacles (obstacles within an obstacle) at any level as well as internal floor plans for a building. Furthermore, in real world we see when a person moves around a corner of an obstacle he does not always keep the same distance from that corner; the anchors also serve this flexibility of realistic movement behavior.

Figure 3. Node movement guided from $P_s$ to $P_d$ by anchors in presence of obstacles with entry point when there is existing pathway.

Figure 4. (a) Anchor $OBXY$ is generated when $90^\circ < 2\theta < 180^\circ$ (b) Splitting of anchor $OBXY$ into two triangular segments $BOX$ and $OXY$ when $2\theta < 90^\circ$.

<table>
<thead>
<tr>
<th>Notation</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>$VG_{ex}$</td>
<td>Set of External General Vertices (EGV)</td>
</tr>
<tr>
<td>$VD_{ex}$</td>
<td>Set of External Door Vertices (EDV)</td>
</tr>
<tr>
<td>$VP$</td>
<td>Set of Path Terminal Vertices (PV)</td>
</tr>
<tr>
<td>$VD_{int}$</td>
<td>Set of Internal Door Vertices (IDV) of obstacle $b$</td>
</tr>
<tr>
<td>$VG_{int}$</td>
<td>Set of Internal General Vertices (IGV) of obstacle $b$</td>
</tr>
<tr>
<td>$B$</td>
<td>Set of obstacles</td>
</tr>
<tr>
<td>$D_b$</td>
<td>Set of doors of obstacle $b$</td>
</tr>
<tr>
<td>$v_o(d_b)$</td>
<td>IDV of door $d$ of obstacle $b$</td>
</tr>
<tr>
<td>$v_e(d_b)$</td>
<td>EDV of door $d$ of obstacle $b$</td>
</tr>
<tr>
<td>$v_i$</td>
<td>Vertex derived from a node’s initial position</td>
</tr>
<tr>
<td>$v_d$</td>
<td>Vertex derived from a node’s destination position</td>
</tr>
<tr>
<td>$N$</td>
<td>Set of nodes</td>
</tr>
</tbody>
</table>

After implementing all the anchors, the graph creation mechanism starts. Usually a vertex is created for each anchor with the corresponding center point $c$. For special cases as shown in Fig. 4(b) if the convex angle $2\pi - 2\theta$ is reflex, i.e. $2\theta < 90^\circ$, the corresponding anchor is divided into two triangular segments $BOX$ and $OXY$ as it is
possible to reach \( P_d \) from \( P_b \), by choosing any intermediate point \( P_{bl} \) and \( P_{dl} \) in each of the respective triangular segments without intersecting any edge \( OD \) and \( OE \) as \( \theta \geq 0 \).

For illustrating the Scenario generation algorithm various notations used are explained in Table 1. Here, by the word “obstacle”, we refer to both accessible (e.g. buildings) and inaccessible (e.g. lakes, hills, restricted area) objects observed in the environment.

### Algorithm 1: Scenario Generator

1. For each obstacle \( b \in B \)
   
   For each door \( d_b \in D_b \), Add edge \( \langle v_{db}(d_b) , v_{vb}(d_b) \rangle \) to edgelist

2. For every pair of vertices \( v_r, v_i \in VG_{ab} \cup VG_{ab} \land v_r \neq v_i \)
   
   if the edge does not intersect with any wall of \( b \) then
   
   Add edge \( \langle v_r, v_i \rangle \) to edgelist and adjustWeight\((v_r, v_i)\)

3. For every pair of vertices \( v_r, v_i \in VG_{ab} \cup VG_{ab} \land v_r \neq v_i \) and both \( v_r, v_i \notin V_p \) and the line \( v_i, v_r \) does not intersect with any wall add edge \( \langle v_i, v_r \rangle \) to edgelist and adjustWeight\((v_i, v_r)\)

4. Add user defined paths with vertices \( v_r, v_i \) to edgelist where \( v_r \in V_p \land v_i \in V_p \land v_r \neq v_i \) and adjustWeight\((v_r, v_i)\)

5. For each \( v \) in \( \{v_r, v_i\} \)
   
   if \( v \) is inside obstacle \( b \) then for each vertex \( v_r \in VG_{ab} \cup VG_{ab} \)
   
   if edge \((v, v_r)\) does not intersect with any wall of \( b \) add to edgelist with “temp” tag and adjustWeight\((v, v_r)\)
   
   else if \( v \) is outside then for each vertex \( v_l \in VG_{ab} \cup VG_{ab} \land V_p \)
   
   if edge \((v, v_l)\) does not intersect with any wall add to edgelist with “temp” tag and adjustWeight\((v, v_l)\)

6. If to reach destination, a node \( n \) wishes to follow user defined path then use user-defined path
   
   else use the path \( n \) wishes to take.

7. Remove edges marked “temp” from edgelist

8. Repeat step 4 to 7 until simulation duration \( T \) expires

9. Repeat steps 4 to 8 for each node \( n \in N \)

Here the \( \text{adjustWeight}(\) function is used to specify the weight (length) of each edge. For the scenario generation purpose we have intuitively reduced the weight to half of the length of edges that represent roads. As a result those edges will get preference in the shortest path computation, and in moving to a destination, a node will most likely prefer an existing path, if any, in most cases as we do in real life. Upon considering the actual phenomena of people’s movement it is flexible enough to select a path based on a particular criteria set by users or any predefined path or a combination of paths which may include part of both shortest and predefined path.

The algorithm returns a set of vertices required to reach the destination from the starting point. Each time it selects a random point inside the next vertex (for special anchors that need to be split into two triangular segments, one random point in each of the segments is used as intermediate destinations as explained in Fig. 4(b)) in the list and continues to do so as long as it reaches the destination without intersecting any wall.

Moreover, we adopted another method for simulating the node movement which does not necessarily follow the shortest path. This movement can be defined as: if \( E = \{e_1, e_2, \ldots, e_j\} \) is the set of edges, for a vertex \( u \) and \( dist(v_1, v_2) \) denotes the distance between vertices \( v_1 \) and \( v_2 \), the next hop will be one of its neighbor (vertices that have an edge with the current vertex) vertices \( v \) for the destination \( d \), if \( \langle u, v \rangle \in E \land \forall v^* \) where \( \langle u, v^* \rangle \in E \land dist(v_1, d) \leq dist(v^*, d). \) When a node reaches a vertex, it selects the vertex among its neighbors which is nearest to destination, similar to packet forwarding method of GPSR. Since there exist obstacles, the next vertex chosen is not necessarily present in the shortest path. Whenever a loop is encountered, another point is selected as destination and the process continues.

For simulating a real event (e.g. emergency relief, rescue operation), there is another feature in AMM which permits the user to specify a number of rectangular sized regions of interest (ROIs). If ROIs exist then a node randomly selects one region and within it a point is selected randomly and tends to move there. After arriving there it again selects a region and this process continues until the simulation time expires. While moving from one region to another, nodes use anchor based mobility model to deal with the environmental context.

Our proposed mobility model is focused to mimic the movement of people in pedestrian friendly area (e.g. university/school campus, techno park) where people can move anywhere except the inaccessible obstacles in the specified area with the freedom and flexibility of using/not using existing pathways. Moreover, it assumes people to be aware of exact position of obstacles, doors and pathways. Mobile environments like City mobility models where one mobile entity’s mobility is affected by other entities, this proposed model may not be suitable in such cases and needs some change. It can be inferred that AMM is a specialized Random Waypoint model that can adopt various geographic attributes. If there is no obstacle and pathway, no anchor is needed to be generated and the model turns into simple random waypoint model.

### 5. Computational complexity

In this section, we present the computational complexity of our proposed mobility model. Finding intersection of two lines and \( \text{adjustWeight}(\) function have constant time complexity. For anchor computation each general anchor is tested whether it intersects with any obstacles. Finding intersection in brute force method of two polygons needs \( O(mn) \) [17] where \( m \) and \( n \) denote the number of corners of each polygon respectively.
Table 2. Notations used in computational complexity analysis.

<table>
<thead>
<tr>
<th>Notation</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>$k$</td>
<td>Number of obstacles</td>
</tr>
<tr>
<td>$p$</td>
<td>Number of IGV</td>
</tr>
<tr>
<td>$q$</td>
<td>Number of EGV</td>
</tr>
<tr>
<td>$e_b$</td>
<td>Number of IGV of obstacle $b$</td>
</tr>
<tr>
<td>$i_b$</td>
<td>Number of doors of obstacle $b$</td>
</tr>
<tr>
<td>$h$</td>
<td>Number of doors</td>
</tr>
<tr>
<td>$f$</td>
<td>Number of PV</td>
</tr>
<tr>
<td>$w_b$</td>
<td>Number of walls of obstacle $b$</td>
</tr>
<tr>
<td>$u$</td>
<td>Number of walls, and</td>
</tr>
<tr>
<td>$r$</td>
<td>Number of general vertices</td>
</tr>
<tr>
<td>$</td>
<td>N</td>
</tr>
<tr>
<td>$T$</td>
<td>Simulation duration</td>
</tr>
<tr>
<td>$I$</td>
<td>Average number of destination selection by a node within $T$</td>
</tr>
</tbody>
</table>

It can be inferred that the total number of corner points of obstacles = the total number of walls = $u$. So, $u = p + q$ as a general anchor is placed at every corner of each obstacle either inside or outside. An IDV and EDV are generated for each door. As a result total number of IDV = total number of EDV = $h$. For our model where anchors have fixed number of (four) corner points the complexity becomes $O(4m) = O(m)$. Resizing of anchors after finding intersection requires constant time. Then the complexity for computing general anchors is:

$$(\sum_{b=1}^{k} O(w_b) ) (\sum_{j=1}^{4} w_j) = O(u^2)$$ since $\sum_{j=1}^{4} w_j = u$. As doorway and pathway anchors are user defined, it is assumed that they will not intersect with any obstacle, making the complexity $O(u^2 + 2h + f) = O(u^2 + h + f)$.

Step 1 in Algorithm 1 generates $O(h) + O\left(\sum_{b=1}^{k} \frac{1}{2}\left(i_b + e_b\right)\right)$ edges and $O\left(\sum_{b=1}^{k} \frac{1}{2}\left(i_b + e_b\right)\right)$ edges need to be checked for intersection with walls of respective obstacles resulting in:

$$O(h) + O\left(\sum_{b=1}^{k} \frac{1}{2}\left(i_b + e_b\right)\right)w_b \leq O\left(h + kw_{\text{max}}\left(i_{\text{max}} + e_{\text{max}}\right)\right)$$

where $w_{\text{max}}$, $i_{\text{max}}$, and $e_{\text{max}}$ are the maximum values of $w_b$, $i_b$, and $e_b$ respectively for all values of $b$.

In step 2, each edge is checked with every wall for intersection requiring:

$$\left(\frac{q + h + f}{2}\right) = \frac{1}{2}\left(f + \sum_{j=1}^{4} w_j\right) \leq O\left(u(q + h + f)^2\right).$$

Step 3 requires $O(r)$.

Step 4 requires either at most $O(2u(q + h + f))$ or at least $O(2w_{\text{max}}(e_{\text{max}} + i_{\text{max}}))$. Since $q \geq e_{\text{max}}$, $h \geq i_{\text{max}}$ and $u \geq w_{\text{max}}$ at worst case it becomes:

$$O(2u(q + h + f)) = O(u(q + h + f)).$$

In step 5, we used shortest path algorithm [18] on the graph with vertices and edges to reach the destination which requires

$$O(\mid V \mid + \mid E \mid) = O\left((p + q + 2h + f + 2)^2 + h + \left(\frac{q + h + f}{2}\right)\right).$$

where, $\mid V \mid = (p + q + 2h + f + 2)$, and

$$\mid E \mid = h + \left(\frac{q + h + f}{2}\right) + \frac{1}{2}\left(i_b + e_b\right) + r.$$

As in the generated graph, $\mid E \mid < \frac{\mid V \mid}{2}$, we can approximate the complexity: $O(\mid V \mid)$

$$= O((p + q + 2h + f + 2)^2) = O((p + q + h + f)^2).$$

Removing temporary edges in step 6 it requires:

$$O(2(q + h + f)) = O(q + h + f).$$

Anchor generation, step 1, 2 and 3 are executed only once which requires:

$$O(\frac{u^2 + kw_{\text{max}}(i_{\text{max}} + e_{\text{max}})^2 + u(q + h + f)^2}{2}).$$

If the average number of destination selection within simulation duration $T$ for one node is $I$ then the total complexity for scenario generation which is loosely bound becomes:

$$O\left(\frac{u^2 + kw_{\text{max}}(i_{\text{max}} + e_{\text{max}})^2 + u(q + h + f)^2}{2} + I \times |N| \times (p + q + h + f)^2\right).$$

On the other hand required complexity for generating Voronoi graph is $O(u \log u)$ [19] since the number of generated Voronoi vertices is $O(u)$. As a result the total complexity for scenario generation in Voronoi model is $O(u \log u + I \times |N| \times u^2)$.

The complexity of our proposed model becomes higher than Voronoi graph based model, because Voronoi model does not take into account the accurate position of doorways and pathways and fails to capture the actual context of environment as explained in Section 3. Whereas AMM model can produce better traces of mobility pattern considering environmental attributes as observed in the real world at the cost of higher computational complexity.

6. Simulation environment

We used NS-2.30 to simulate the proposed and existing mobility models. The network scenario is generated based on a real context, the locations of buildings of the Clayton campus of Monash University, Australia as shown in Fig. 5. We selected a portion in the map keeping the playground size 1000m × 1000m and within this terrain the existing pathways, buildings and
actual position of main entrances (doors) to buildings were incorporated as shown in Fig. 6.

Figure 5. Snapshot of a portion of Clayton campus (courtesy: Google Earth).

150 nodes were used in simulation where 95% nodes follow the AMM algorithm, while the rest do not use shortest path. The side of the general and doorway anchors was 20 meters and 5 meters respectively. We experimented at 4 different maximum speeds of 5, 10, 15 and 20 (m/sec) and performed 10 runs with different mobility scenes and averaged results are presented here.

The duration of simulation was 900 seconds. In each second 15 pairs of nodes were randomly selected, each pair consisted of one sender and one receiver and 128 byte packet was sent to the receiver. When a node reached the destination, it paused there for 1 second and then randomly selected another destination. For routing data packets GPSR protocol was used while IEEE 802.11 protocol was used at the MAC layer. GPSR uses greedy forwarding, where a node forwards packets to a neighboring node that is closest to the destination among all neighbors. In situations where such a greedy path does not exist (i.e., a node cannot find a neighbor which is closer to destination than itself), GPSR recovers by forwarding in perimeter mode, in which a packet traverses successively closer faces of a planar subgraph of the full radio network connectivity graph, until reaching a node closer to the destination, where greedy forwarding resumes.

In order to simulate situations possibly occurring in real context, a wide variety of simulation scenarios was developed by the AMM model and experimented with. However, considering the space limitation, in this paper, we present five variations: 1) InaccObs: for representing inaccessible obstacles, nodes are not allowed to enter the buildings and there is no ROI and pathway, 2) ROI: obstacles are accessible through doors and there are ROIs but no pathway, 3) NoROI-NoPath: obstacles are accessible through doors and there are ROIs but no pathway, 4) NoROI-Path100: obstacles are accessible through doors and 100% times nodes use user defined pathways and 5) NoROI-Path50: obstacles are accessible through doors and 50% times nodes use user defined pathways. We used 3 rectangular shaped ROIs highlighted with dotted lines (A, B, C in Fig. 6).

The visualization of a node movement is presented in Fig. 7. A node starts from $s_0$ at $t_0$, progressively selects $d_{p_1}$, $d_{p_2}$, $d_{p_3}$, $d_{p_4}$ at $t_1$, $t_2$, $t_3$, $t_4$ respectively and finally ends at $d_p$ at $t_6$ when the simulation duration expires, where $t_0 < t_1 < t_2 < t_3 < t_4 < t_5 < t_6$. For comparison we selected widely used Random Direction model [5], and Voronoi graph based obstacle mobility model [1]. Since the assumption made in [1] that there exist a door in the wall where the Voronoi edge intersects did not actually happen all the time, we trimmed edges residing in the buildings that do not coincide with the doors, to find out how Voronoi based model would perform in real world door placement.

Figure 6. A snapshot of MobiGen, modeling Clayton campus (Buildings, pathways and doors).

Figure 7. A snapshot of MobiGen tracing a node’s movement path.
7. Results

Various network parameters including average number of neighbors at different time span and speed, average link duration, average path length in terms of hops and packet delivery success rate were calculated. These measures are also investigated in simulation of similar studies [1, 7]. To investigate the impact of radio transmission range, we chose a number of transmission ranges with TwoRayGround propagation model as defined in NS-2.30. For space limitations, we present results for only 150m in this paper though other transmission ranges produced similar trend.

![Figure 8. Variation in average number of neighbors over time (at 10m/s).](image)

Fig. 8 shows average number of neighbors at different time stamp throughout the simulation time. From the figure it can be concluded that the mobility models actually fall into three groups: 1) Random direction and Voronoi that distribute nodes sparsely throughout the simulation area, 2) ROI model that tends to concentrate the nodes in particular geographic regions and 3) The other models (NoROI-NoPath, NoROI-Path50, NoROI-Path100 and InaccObs) that fall in between. Due to different level of concentration caused by different models, average number of neighbors (Fig. 9) and durability of link with respect to node movement speed (Fig. 10) are affected. Since in ROI nodes are very much concentrated in specific regions the average number of neighbors and link duration is the highest. While average number of neighbors remains unaffected with node movement speed, link duration decreases significantly with increasing speed, irrespective of mobility models. Number of neighbors and link duration has direct impact on average number of hops for packets from sender to receiver (Fig. 11). Here it is seen that the models belonging to Group 1 result in higher path length than any other groups and Group 2 results in the lowest due to different level of node concentration. Finally, the path length has direct impact on packet delivery success rate (Fig. 12) as the models having less path length show higher success rate.

![Figure 9. Average number of neighbors](image)

In Fig. 8, Random direction model and Voronoi model show almost the same number of neighbors during the simulation duration. On the other hand, other models show an increase in number of neighbors within the first 150 seconds. Voronoi model only permits to place nodes on the Voronoi graph vertices and edges resulting in almost uniform distribution of neighbors over the simulation time. In all variations of AMM at the initial stage nodes are randomly placed in the simulation area, but as time progresses, their movement pattern becomes more restricted yielding an increasing trend in the number of neighbors during the initial stage of simulation.

![Figure 10. Average link duration](image)

![Figure 11. Average path length from packet source to destination](image)

Fig. 9 shows the average number of neighbors at different speed. When the ROIs are present, nodes travel from one region to another and if possible use doorways to go through buildings. This makes node movement
much concentrated on the interconnection of the regions and higher number of neighbors is available.

As illustrated in Fig. 10, Random direction model experiences the lowest link duration and it is justified by the nature of unrestricted node movement in any direction. Voronoi model has higher link duration than InaccObs, NoROI-NoPath, NoROI-Path100 & NoROI-Path50 at higher node speed, by limiting node movement to the Voronoi edges, the effective area where the nodes are available becomes less resulting in higher clustering of nodes and thus the link duration increases. On the other hand, ROI shows the highest link duration. Since there are 3 ROIs in the simulation scenario (Fig. 6), it is almost 33.33% chance that a node will select the same region, where it resides in. However this link duration as well as packet transfer success rate, path length, average neighbors of ROI are subject to be affected by the number, placement and size of ROIs.

Fig. 11 shows the average path length from sender to receiver in terms of hops. Though average number of neighbors in Voronoi is higher than Random direction model, it has higher path length from sender to receiver. For GPSR having sufficient neighbors alone is not enough, rather existence of nodes in the direction of receiver is important to forward packet in greedy mode. As the nodes in Voronoi model moves on the voronoi edges only, because of the unavailability of nodes in the Voronoi cells, nodes don’t find enough nodes in the direction of receiver. Consequently they fail to find shorter routes (for switching to perimeter mode more frequently) and often fall into routing loop due to the nature of GPSR protocol. Other than InaccObs, in all variations of AMM, since nodes are permitted to enter the buildings through doors only, a few nodes are expected to be present there, and these nodes act as some sort of gateway between two regions separated by an obstacle. This fact results in lower path length and ROI having the lowest as expected.

Fig. 12 shows packet delivery success rate at different speed of node movement. Results show that, in Voronoi based model, nodes are only permitted to travel on the graph edges resulting in the lowest success rate as packets have to travel longer path as observed in Fig 11. Moreover due to trimming of some edges that do not coincide with existing doors of obstacles results in larger empty space, and for this reason packets need to select longer path and consequently packet loss increases. Random direction model shows better success rate than Voronoi model, since nodes are totally scattered throughout the simulation area, but not restricted to any path, yielding relatively shorter paths for being able to find nodes in the direction to receiver. For InaccObs where nodes can not enter buildings, the success rate is less than any variations of AMM, because absence of nodes inside buildings causes packets to travel around the buildings and longer path creates more failure in packet delivery. For NoROI-Path100 nodes are concentrated on the predefined pathways and since the number of defined roads is less than that of Voronoi graph, most of the packets are sent to nearby nodes, resulting in higher success rate than Voronoi model. The success rate of NoROI-Path50, which is actually a combination of NoROI-NoPath and NoROI-Path100, stays in between as expected, since 50% of time nodes chose to use roads. The presence of nodes in obstacles reduces perimeter mode forwarding in NoROI-NoPath model, resulting in higher success rate. In ROI where nodes move on a region of interest basis, the success rate is the highest, since nodes are concentrating in ROIs and their interconnecting areas. At the end it can be inferred that while using models belonging to Group 2, the routing protocol has the highest success rate, where as with Group 1 it performs lower, and for Group 3 the success rate stays in between.

To test the significance of difference in success rate (Fig. 12) we selected NoROI-NoPath and Random direction as representatives of the proposed AMM models and existing models, respectively. The t-test at speed 5, 10, 15 and 20 m/sec yielded p-values of p < 3.18×10^{-4}, 2.59×10^{-6}, 2.10×10^{-8}, 2.38×10^{-7} at 99% confidence level, validating their performance difference being statistically significant.

**Figure 12.** Packet delivery success rate

From the analysis on different metrics stated above the following conclusion can be drawn:

- Existence of ROI makes significant difference in performance evaluation of routing protocol
- Existence of accessible obstacles makes a medium level difference compared to inaccessible obstacles.
- Path preference details make a small difference.

It should be noted that the simulation result only demonstrates the impact of our mobility model on the routing protocol rather than evaluating the protocol itself. The results show that there exist significant impact of the choice of mobility model on routing protocol performance and ignoring or approximating the environmental attributes by other existing models will
lead to less accurate result which has been taken care of in the proposed model.

8. Conclusion

The success and efficacy of simulation environment to evaluate MANET’s performance highly depend on how closely the mobility of nodes resembles real-world scenario which existing models fall short of. In this paper, we propose a new mobility model along with a scenario generation tool for MANET that enables to create a wide variety of simulation scenario closely resembling to real world environment. The characteristics and impact of the proposed mobility model in the presence of obstacles and pathways at different movement patterns are analyzed in details. A comparison of the characteristics of the proposed model with other existing ones reveals the significant impact of the choice of mobility model on performance evaluation. Since the generated scenario can be easily plugged in Network Simulator NS, this scenario generator MobiGen will be helpful for other researchers in MANET and wireless communication in general. Currently the model assumes unobstructed omni-directional radio; however, it can be further extended to incorporate fading effect due to the presence of obstacles considering their material composition. Further study using a wide range of scenarios with different traffic patterns, radio model and statistical evaluation is needed to comprehend the actual effects of the proposed model on the overall performance of the routing protocol and remains the focus of our future study.

Acknowledgement

The authors would like to formally thank Carey Williamson for shepherding this paper and all anonymous reviewers and program committee members for their constructive and perceptive comments, suggestions and criticisms, and believe the paper has undoubtedly benefited significantly from the amendments recommended.

References


