A Method for Custom Movement Generation in Wireless Mobile Network Simulation

Hawra Alseef, John DeDourek, Przemyslaw Pochec
Faculty of Computer Science
University of New Brunswick
Fredericton, Canada
e-mail: {hawra.alseef, dedourek, pochec}@unb.ca

***Abstract*—Random motion is often used in evaluating performance of Mobile Ad hoc Networks (MANETs). Mobility pattern of nodes significantly affects performance of MANETs. In the simulation of large mobile networks, automated movement generators are used, such as the *setdest* utility in ns2. In this paper, we investigate the modifications to the standard *setdest* generator that specifies the motions along straight-line paths. We propose and implement a new method for movement generation for the ns2 simulator that specifies the node movements along curved paths generated using simple fractals. The new generator was successfully tested with the ns2 simulator. The results show that, in the random way point motion, only the node speed significantly affects MANET performance, and not the shape of the individual path segments taken by a node.**

Keywords-movement generator; network simulation; ns2; fractal path; setdest utility; MANET

I. INTRODUCTION

A Mobile Ad hoc Network (MANET) is a set of mobile devices that cooperate with each other by exchanging messages and forwarding data [1][2]. Mobile devices are linked together through wireless connections without infrastructure and can change locations and reconfigure network connections. During the lifetime of the network, nodes are free to move around within the network and node mobility plays a very important role in mobile ad hoc network performance. Mobility of mobile nodes significantly affects the performance of a MANET [2].

Simulation is a commonly used evaluation tool for mobile networks. It allows for modelling existing networks as we as future networks. Using simulation, different network configurations working under different traffic load conditions and using different routing protocols can be quickly and easily modelled and evaluated. For mobile networks, if no other design constraints are present, random motion of the nodes is usually used. Using a common and specific random motion model allows to create the base condition for comparison between different network evaluations.

ns2 is an open source simulator well suited for modelling wired and wireless networks [3]. It includes a motion scenario generator *setdest* designed to automatically generate random motion paths for a large number of nodes. This tool generates a random motion path for each node by selecting a random destination for the node and then moving the node towards this destination along a straight line. Once this destination is reached by the node, a new destination is randomly selected and, after an optional pause time, the node starts moving again to the new destination.

In this paper, we propose to use the random motion generated by the ns2 *setdest* utility [3] to create a new trajectory for the mobile nodes. The waypoints are kept the same but the path followed by the node between two waypoints is no longer defined by one straight line segment, and is replaced by a fractal curve composed of a number of shorter line segments.

In Section 2, we review different random movement models commonly used in simulation. Section 3 introduces properties of the Koch fractal. Section 4 describes the new movement generator based on generation of the node movement along a fractal path. Section 5 presents a study of two mobile networks: one with the conventional random movement and the other with the fractal movement. Conclusion is presented in Section 6.

II. STATE OF THE ART

Any model of a MANET requires a mobility model specifying the movement pattern of the nodes [4]. The most realistic models are trace driven but cannot be always applied because of their *a posteriori* nature. On the other hand, the synthetic models [5] are not trace driven but instead rely on assumptions about the node movement mode. Among these are the random (random-based) models where the nodes move randomly and without restrictions and where the destination and the speed are chosen randomly.

There are many different types of random mobility models that are used in MANETs. The main ones are the Random Walk, the Random Waypoint, and the Random Direction. The Random Walk model [5] mimics the Brownian motion of particles found in nature. Each node travels in a straight direction for a specified time interval before randomly changing the speed and the direction, and then continuing for another time interval. In the Random Waypoint model [6], each node selects a destination within the simulation area and then follows a straight path to it; once the destination is reached the node may pause and then select a new destination (waypoint). In the Random Direction model [7], instead of selecting a random destination, the node selects a random direction and then moves along this

 ISBN: 978-1-61208-422-0

direction until it reaches the simulation area boundary where, possibly after a pause, it selects a new direction for the next move.

The ns2 *setdest* utility generates the node movements following the Random Waypoint algorithm [3]. In the Random Waypoint Movement (RWP), each node moves from its randomly selected initial starting position towards the randomly selected at a randomly selected speed. Once at the target destination the node may pause for a randomly selected time, and then start the next random move. This process will be repeated until the end of the simulation by the ns2. One notable aspect of the RWP movement is that the nodes following this pattern tend to concentrate in the center region of the deployment area [8][9].

III. THE KOCH FRACTAL

We propose to use fractals for the movement generation for mobile network simulation based on the RWP model. Instead of moving the nodes along a straight line between the waypoints, the nodes are moved along a fractal path. We selected the Koch fractal because, like a line segment, it has a defined starting and ending points, and because of the simplicity of its generating algorithm [10][11].

A. Construction of the Koch curve

The construction of the Koch starts with a straight line that is then converted to the Koch fractal curve, Figure 1.

Figure 1. Step 1.

Figure 2. Step 2.

Figure 3. Step 3.

This process is then repeated for each of the 4 segments generated at the first iteration, leading to the curve shown in Figure 3. These steps can be applied repeatedly and eventually result in a complex shape. When the Koch curve generating algorithm is applied to an equilateral triangle it results in a closed curve called the Koch snowflake [11].

B. Properties of the Koch snowflake

Number of Sides (n): for each iteration, every segment of the curve from the previous iteration will be converted to four segments in the following iteration. Since we begin with three sides, the formula for the number of sides in the Koch curve is:

$$n = 3 * 4^a \quad (1)$$

where a indicates the number of iterations. For iterations 0, 1, 2, and 3, the numbers of sides are 3, 12, 48, and 192 respectively.

Length of Sides (L): In every iteration, the length of a side is 1/3 the length of a side from the previous iteration. If we begin with an equilateral triangle with side length x, then the length of a side in iteration a is:

$$L = x * 3^{-a} \quad (2)$$

For iterations 0 to 3, length = x, x/3, x/9, and x/27.

Perimeter (p): The key features of the Koch curve lies in having the same length of all sides in each iteration, this leads to a perimeter, which is simply the number of sides multiplied by the length of a side:

$$p = n * L \quad (3)$$

For the snowflake, from the previous formulas, we get:

$$p = (3 * 4^a) * (x * 3^{-a}) \quad (4)$$

In the same manner, for the first 4 iterations (0 to 3) the perimeter is 3x, 4x, 16x/3, and 64x/9. We notice that, the perimeter increases by 4/3 times for each iteration, so we can rewrite the formula as

$$p = (4/3)^a * 3x \quad (5)$$

IV. CUSTOM MOVEMENT GENERATION WITH FRACTALS

The main objective of this research is to implement a new method for movement generation in MANET simulation in ns2. Indeed, the standard way for movement generation is to use the *setdest* utility that generates a set of setdest commands that are then "executed" in the ns2 simulator. setdest commands generate a movement along a straight line between the current location and the designated destination point. This research aims at providing a new tool for modifying the simulation environment by modeling motion in wireless network simulations, specifically for generating movement files for ns2 simulation that specify the motion along curved (fractal) paths. Typically, defining the node movements needs to be done ahead of the ns2 simulation. In general a curved path can be approximated by a series of short line segments, which determine the final shape of the curve. Therefore, a Java program was implemented that reads the

 ISBN: 978-1-61208-422-0

movement file with random movements generated, for example, by the *setdest* utility. Then, as each movement in the movement file is specified by a separate setdest command, we will replace each one of these setdest commands, each specifying a movement along a straight line, with a series of setdest commands specifying the movement along a curved path (fractal). Once the new movement file is generated the ns2 simulation can proceed in a standard way.

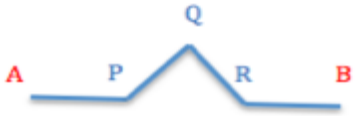

Figure 4. The result of fractal transformation of a line segment AB.

Let's consider the original setdest command for the direct movement from A to B (Figure 4):

$ns_ at T "$node_(#) setdest X_B Y_B S"

where T indicates the starting time at which the node starts moving towards the destination X_B,Y_B at the specified speed S. While splitting the initial path (line segment AB) into four segments (AP, PQ, QR and RB) and defining the destination of each of the four moves is a simple geometry, the other setdest command parameters require careful consideration. More precisely, the need of updating the time and speed in the setdest commands arises when applying the fractal transformation. In order to make the fractal movements arrive at the final destination (point B) at the same time that the original straight movement would have arrived, we need to do the following modifications:

$ns_ at T_P "$node_(#) setdest X_P Y_P S_{new} "
$ns_ at T_Q "$node_(#) setdest X_Q Y_Q S_{new} "
$ns_ at T_R "$node_(#) setdest X_R Y_R S_{new} "
$ns_ at T_B "$node_(#) setdest X_B Y_B S_{new} "

The four (fractal) movements should proceed sequentially, each having a starting time after the previous movement ends. To calculate the precise time of each move and the new speed we need to determine the new speed and the new starting time for each of the four new setdest commands. First, we need to calculate the time the node would take to travel from A to B at speed S along the original straight line path AB:

$$t_{AB} = \text{sqrt}((X_B - X_A)^2+(Y_B-Y_A)^2)/S \quad (6)$$

then the start times for each move are calculated as:

$$\begin{aligned} T_P &= T \\ T_Q &= T_P + t_{AB}/4 \\ T_R &= T_Q + t_{AB}/4 \\ T_B &= T_R + t_{AB}/4 \end{aligned} \quad (7)$$

and the new speed, due to the distance travelled increased by 1/3, is:

$$S_{new} = 4 * S / 3 \quad (8)$$

(Obviously, when the intermediate point Q would fall outside the predefined simulation region then the corresponding segment of the fractal path cannot be generated, as shown on Figures 5 and 6.)

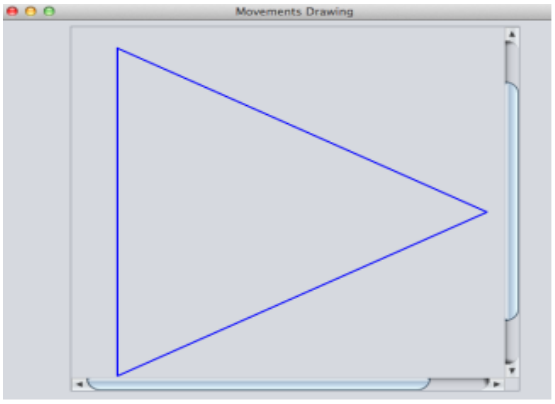

Figure 5. Trace of sample simple node movement.

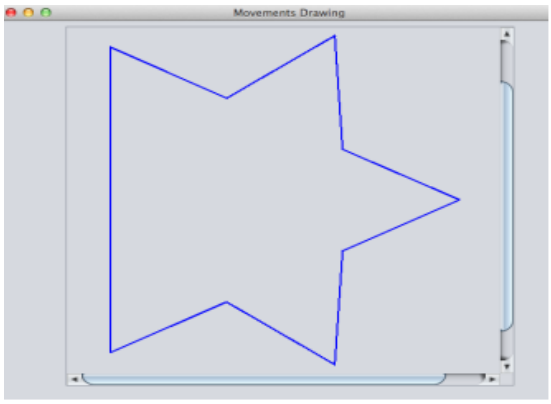

Figure 6. Fractal movement generated from Figure 5.

For example, consider the following movement statement taken from a movement file generated by the setdest utility:

$ns_ at 0.400000 "$node_(0) setdest 100.00000000 400.0000000 1000.02523710421"

This line specifies that at time 0.40000s, node0 starts to move from the starting point (100,100) towards the destination (100,400) at a speed of 1000m/s (this can be one single random movement in a straight line). This single command in the movement file is then replaced by four new commands generating the movement along the path corresponding to the shape of the Koch fractal (one iteration of the Koch fractal generation algorithm). The four movements are listed below.

 ISBN: 978-1-61208-422-0

$ns_ at 0.40000000 "$node_(0) setdest 100.00000 200.00000 1333.366982805613"

$ns_ at 0.474998107265 "$node_(0) setdest 13.3974596215 250.0000 1333.36698280"

$ns_ at 0.549996214530 "$node_(0) setdest 100.00000 300.00000 1333.366982805613"

$ns_ at 0.624994321795 "$node_(0) setdest 100.00000 400.00000 1333.366982805613"

V. EVALUATION OF MANET PERFORMANCE UNDER FRACTAL MOVEMENT

We evaluated the performance of a sample MANET under different motion generation conditions. A MANET with the number of nodes ranging from 5 to 80 was simulated over the area of 800 by 800 meters with two fixed communicating stations at (100,500) and (700, 500). Constant Bit Rate (CBR) traffic was generated over the User Datagram Protocol (UDP) and routed with Ad hoc On-Demand Distance Vector (AODV) (Table I).

TABLE I. SIMULATION PARAMETERS

Parameters	
Simulator	NS-2.33
Channel Type	Channel / Wireless Channel
Network Interface Type	Phy/WirelessPhy
Mac Type	Mac/802.11
Radio-Propagation Type	Propagation/Two-ray ground
Interface Queue Type	Queue/Drop Tail
Link Layer Type	LL
Antenna	Antenna/Omni Antenna
Maximum Packet in ifq	50
Area (n * n)	800 x 800
Source Type	(UDP) CBR
Simulation Time	100s
Routing Protocol	AODV

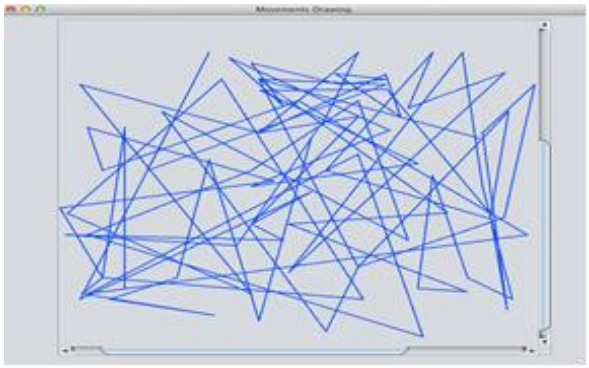

Figure 7. Trace of complex random movement.

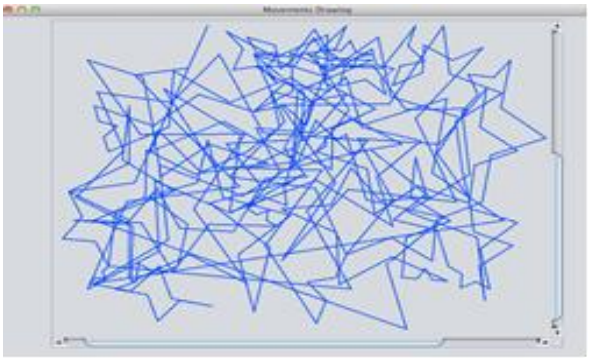

Figure 8. Fractal movement generated from Figure 7.

Standard RWP movement was generated with setdest and then the standard movement was converted to the fractal movement using one step of Koch generating algorithm, as shown in Figures 7 and 8. Two scenarios were investigated: (i) low speed (10m/s), and (ii) high speed (30m/s).

Figure 9 illustrates the difference in the number of packets received at the destination when using the original movement and the new fractal movement at low speed. It shows that most of the time the packet delivery for the fractal movement is higher than the original linear movement. Although the speed of the fractal path is higher than the original (because of the increased path length along the fractal curve between the original waypoints), we observed a higher number of packets delivered at the destination for the fractal movement at speed of 13m/s. However, applying the t-test for the comparison of two paired means representing the packets received in the linear motion and the fractal motion experiments with 25 nodes gives 8%, which indicates that the observed difference is not statistically significant. Also, comparing the average packet delivery across all node densities does not show a significant difference (t-test value 49%).

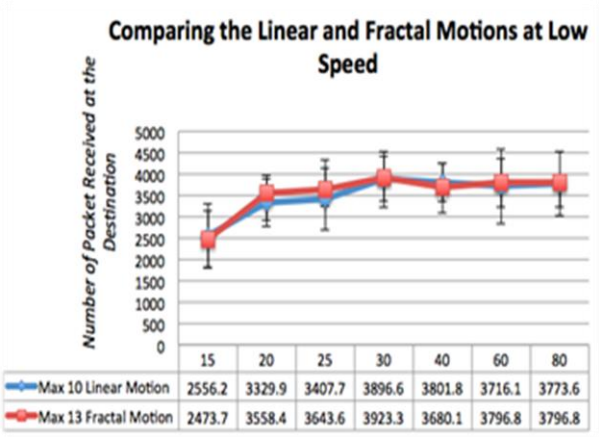

Figure 9. Throughput comparison at low speed

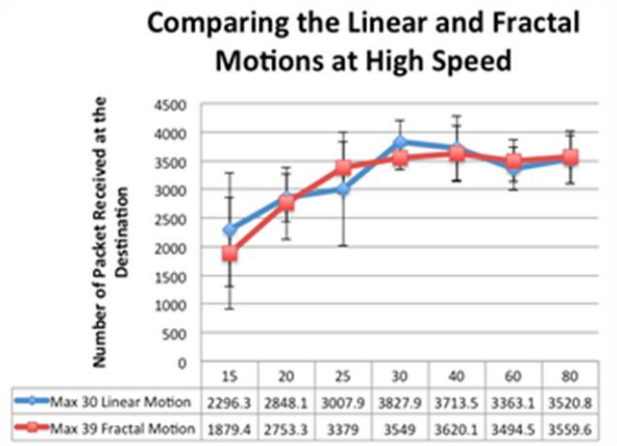

Figure 10. Throughput comparison at high speed.

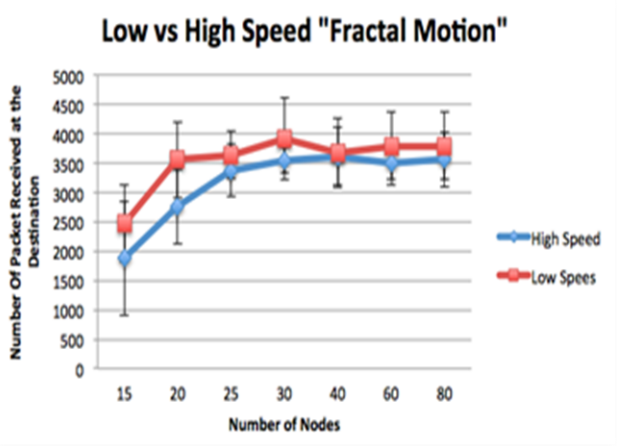

Figure 12. Throughput comparison for fractal motion.

Figure 10 shows the packet delivery for linear and fractal motions at high speed. This time we observe a lower packet delivery for fractal motion recorded in most of the experiments. One possible explanation of lower performance with fractal motion is that the increase in movement speed of 10m/s, from 30 to 40, results in more frequent link disconnections and consequently lower packet delivery. Applying the t-test for the comparison of two paired means representing the packets received in the linear motion and the fractal motion experiments with 25 nodes gives 32%, which indicates that the observed difference is not statistically significant. Also, comparing the average packet delivery across all node densities does not show a significant difference (t-test value 60%).

Figure 11 illustrates the advantage of using lower speed in a network with linear motion. It shows that the packet delivery is consistently higher at low speed for almost all node densities. Applying the t-test for the comparison of two means representing the packets received in the linear motion and the fractal motion experiments with 20 nodes gives 4%, which indicates that the observed difference is statistically significant. The average packet delivery for all node densities is 3176 at high speed and 3497 at low speed,

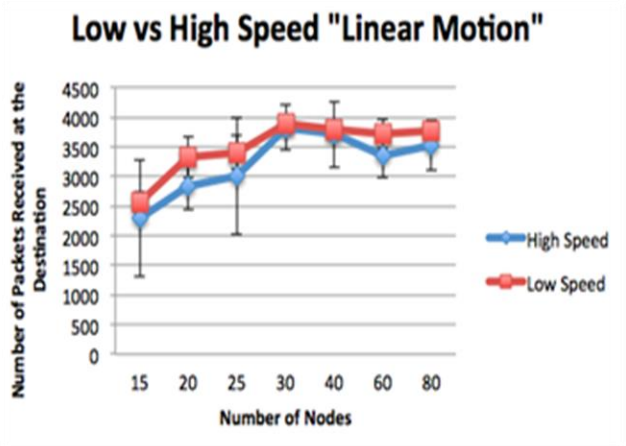

Figure 11. Throughput comparison for linear motion

and this difference in performance is statistically significant (t-test value 0.076%).

Figure 12 illustrates the advantage of using lower speed in a network with the fractal motion. The packet delivery is consistently higher at low speed for all node densities. Applying the t-test for the comparison of two means representing the packets received in the linear motion and the fractal motion experiments with 20 nodes gives 1%, which indicates that the observed difference is statistically significant. The average packet delivery for all node densities is 3186 at high speed and 3553 at low speed, and this difference in performance is statistically significant (t-test value 1.7%).

V. CONCLUSION AND FUTURE WORK

In this paper, we presented a tool for transforming linear movements into fractal movements based on the Koch curve. The new tool reads a standard ns2 movement file, decodes each movement, and replaces it with a series of new movements forming a fractal curve, and then outputs a new movement file. The newly generated movement file satisfies the ns2 specifications and can be used in the ns2 simulator. Both standard movement files generated with *setdest* and new movement files generated with the new fractal tool were used in simulating a MANET with varying number of nodes (i.e. with different node densities). We compared the MANET performance in terms of packet delivery under two different motion scenarios and at different speeds. We observed marginally higher performance of MANET with fractal motion at low movement speeds. However, the statistical tests show that the difference observed in our limited experiments is not significant. We observed that the packet delivery is lower at higher speeds for both motion types, and after the application of the t-test for the difference of the means, we concluded that the observed lower packet delivery at higher speed is statistically significant.

From our results, we conclude that only the node speed significantly affects the MANET performance, and not the shape of the path taken by a node.

 ISBN: 978-1-61208-422-0

The work presented in this paper demonstrated a new experimental approach for investigating performance of mobile networks: applying transformations to the node movement paths. The future work on transforming the node movement paths will include using more than on iteration of the generating function of the Koch fractal, calibrating the node speed when it starts moving on the new curved path and testing if the new path generators reduce the tendency observed in the RWP model of clustering the nodes towards the center of the experimental area.

ACKNOWLEDGMENT

This work is sponsored and funded by the Ministry of Higher Education of Saudi Arabia through the Saudi Arabian Cultural Bureau in Canada.

REFERENCES

[1] S. Basagni, M. Conti, S. Giordano, and I. Stojmenovic (Eds.), Mobile Ad Hoc Networking, New York, Wiley-IEEE Press, 2001.

[2] F. Bei and A. Helmy, A survey of mobility models in wireless Ad hoc Networks, University of California, USA, 2004.

[3] H. Ekram and T. Issariyakul, Introduction to Network Simulator NS2, Springer, 2009.

[4] N. Aschenbruck, E. G. Padilla, and P. Martini, "A survey on mobility models for performance analysis in tactical mobile networks", Journal of Telecommunications and Information Technology, vol. 2, 2008, pp. 54-61.

[5] T. Camp, J. Boleng, and V. Davies, "A Survey of Mobility Models for Ad Hoc Network Research", Wireless Communication and Mobile Computing (WCMC): Special issue on Mobile Ad Hoc Networking: Research, Trends and Applications , vol. 2, no. 5, 2002, pp. 483-502.

[6] J. Broch, D. A. Maltz, D. B. Johnson, Y.-C. Hu, and J. Jetcheva, "A performance comparison of multi-hop wireless ad hoc network routing protocols", Proceedings of the Fourth Annual ACM/IEEE International Conference on Mobile Computing and Networking (Mobicom98), ACM, October 1998, pp. 85-97.

[7] E. M. Royer, P. M. Melliar-Smith, and L. E. Moser, "An Analysis of the Optimum Node Density for Ad hoc Mobile Networks", Proceedings of the IEEE International Conference on Communications(ICC), Helsinki, Finland, June 2001, pp. 857-861.

[8] C. Bettstetter, "Mobility Modeling in Wireless Networks: Categorization, Smooth Movement, and Border Effects", ACM Mobile Computing and Communications Review, vol. 5, no. 3, July 2001, pp. 55-67.

[9] R. Alghamdi, Movement Generator for Mobile Network Simulation. Master's Report, Faculty of Computer Science, University of New Brunswick, Fredericton, Canada, 2012.

[10] G. Edgar, Measure, Topology, and Fractal Geometry, 2nd Ed., Springer 2008

[11] Koch's Snowflake, online, http://en.wikipedia.org/wiki/Koch_snowflake, retrieved: May 2015.

 ISBN: 978-1-61208-422-0