Avoiding Border Effect in Mobile Network Simulation

Raid Alghamdi, John DeDourek, Przemyslaw Pochec
Faculty of Computer Science
University of New Brunswick
Fredericton, Canada
Email: r.alghamdi, dedourek, pochec@unb.ca

***Abstract*—Simulation of mobile networks requires reliable movement generation. Random movement pattern is frequently used in simulators. Standard movement generator setdest in ns2 suffers from border effect, i.e., shows bias towards placing the nodes in the center of the simulated area. We propose and implement a different method for random movement generation in ns2 simulator based on the boundless movement mobility model. By using the quadrats count statistical testing, we show that our movement generator improves the randomness of the node distribution during the simulation.**

Keywords- movement generator; network simulation; setdest utility; MANET; VMR; Quadrats Count; ns2.

I. INTRODUCTION

Communication networks are divide into two main categories: wired and wireless. Wired networks exist between a number of devices connected to each other using connecting media, such as cables and routers. Wired networks can be applied within an area limited by the cables and routers that allow for sending and receiving of data. Wireless networks, on the other hand, are free of such space limitations, and are more easily able to connect different devices to each other. Wireless nodes can play the roles of both hosts and routers, which forward the packets to neighboring nodes.

The mobile ad hoc network (MANET) [1] is a sub-category of ad hoc networks. With the advent of newer technologies, mobile ad hoc networks are becoming an integral part of next-generation networks because of their flexibility, autocongfuration capability, lack of infrastructure, ease of maintenance, self-administration capabilities, and cost-effectiveness [2]. A MANET contains mobile nodes that can be connected wirelessly to each other, for example through either Wi-Fi or Bluetooth. Nodes can be connected over wireless links in ad hoc fashion without central control; this is one of the main advantages of MANETs. In addition, MANET is dynamic and does not rely on fixed or static structure. Consequently, the frequent changes that occur in network topology impact mobile ad hoc network protocols' performance [2]. Because of this very dynamic structure, designing a new MANET often relies on a simulation modeling. In turn, simulation requires a reliable movement generation. In this paper we propose an improvement to setdest, the existing and popular ns2 utility.

The structure of this paper is as follows. In Section II, we discus different movement models for MANETs, including the setdest utility used in ns2. In Section III, we introduce performance measures for evaluating a movement generator. In Section IV, we present a new movement generator. The performance of the new generator is discussed in Section V.

II. MOVEMENT TYPES USED IN MANETS

The mobility models for MANETs can be grouped into two categories: random movement models (which will be discussed late in this paper) and uniform movement models. The uniform movement models include four well-known models: Boundless Simulation Area Mobility Model, Gauss-Markov Mobility Model, A Probabilistic Version of the Random Walk Mobility Model, and City Section Mobility Model [9]. First, a boundless simulation area model is based on the velocity of the mobile node of the current direction and the previous direction [4]. A Gauss-Markov Mobility Model resembles a random model but in fact it is not because it follows a pattern that could be calculated in advance [9]. The Gauss-Markov model is calculating two main parameters of each mobile node which are speed and direction at a certain time instance based on last instances update [9][18][19]. The Probabilistic Version of the Random Walk Mobility Model is a model that uses the probability to determine the next position by using the node state at each position [9][20]. The probability of the Probabilistic model is going higher when the mobile node keeps following the previous direction and is lower if the direction is to be changed [9]. Finally, the City Section Mobility Model is a realistic movement model where the movement of the nodes is still random but the paths are constrained to the grid representing streets in a city [8][9].

A. Random Movement

The Random mobility models used in MANETs differ in the way the nodes move. These principal movement types are: Random Walk, Random Waypoint and Random Direction. Each mobile node in Random Walk has a randomly generated starting position. The nodes travel from their starting position to a randomly generated new location by generating random direction and velocity [9]. A node changes direction and speed either at the end of a time

 ISBN: 978-1-61208-245-5

interval t or if it traveled a distance d. In Random Waypoint, the movement is not constant whereas pause times are introduced. The nodes start at randomly chosen positions, then ”pause” for some time and then start moving at a random velocity towards a chosen destination. The nodes in this model have to ”pause” for some time before they change direction or speed [9]. The drawback in Random Waypoint is clustering near the center (i.e. having the nodes near each other near the center of the experimental area). Random Direction Mobility model was introduced to overcome this drawback in the Random Waypoint model. In Random Direction model, a node travels at a chosen velocity in a chosen direction until it reaches the boundaries of the area rather that until it reaches a randomly chosen location. Once a node hits the boundaries it pauses for a time t and chooses a new direction and starts moving in this new direction again, and so on [7]. The direction is changed only when a node hits a boundary.

B. ns2 Setdest Utility

Setdest is a tool used to generate nodes movements for the mobile nodes in the network simulation ns2 by positioning network nodes in a bounded area and setting the movement in a random direction [10][11]. Setdest tool (version v2) uses the random waypoint mobility model algorithm to create the random movements for the mobile nodes [10][11][12][21] and accept the following parameters: number of nodes, maximum speed, minimum speed, speed type, pause time type, simulation time, x coordinate, and y coordinate) [15][22].

III. INVESTIGATING MOVEMENT GENERATOR PERFORMANCE

We tested the Setdest utility and our new movement generator in two ways: by evaluating the randomness of the position of mobile nodes at different times in the course of simulation, and by comparing the delivery ratio in a simulated MANET in different regions of the experimental area.

A. Randomness Performance

For testing the randomness performance we used the Quadrats Count methodology and the Variance to Mean Ratio (VMR). The Quadrats Counts methodology is an established technique used for analyzing spatial point patterns present in an area by dividing this area into a certain number of sub-areas and then counting the number of points in each sub-area independently [16]. The Variance to Mean Ratio is a statistical test that describes a spatial distribution. We calculate the mean $\bar{x}$, and the variance s^2, of the number of points (network nodes) in each subarea in the Quadrats Count method. The closer the ratio

$$VMR = \frac{s^2}{\bar{x}} \tag{1}$$

is to 1, the more the points are randomly distributed.

B. Delivery Ratio

We tested the boundary effect (i.e., changes in nodes density along the edges of experimental area) by using a random movement generator to run the MANET simulation and then transmitting the packets through the center and also transmitting the packets along the edges. We used a CBR (constant bit rate) [23] traffic over UDP [24], using AODV routing [25], and counted the total number of bytes delivered at the destination node. The higher the delivery ratio for transmission along the edges, the better the movement generator.

IV. CUSTOM JAVA MOVEMENT GENERATOR

The setdest utility uses the random waypoint mobility model algorithm. The waypoint algorithm is known to have the border effect, which creates a kind of clustering at the center of the simulation area [7]. The border effect problem can be avoided by following a different algorithm called the boundless simulation area mobility model. The main idea of the boundless model is to allow going over the edges thus avoiding the influence of the simulation areas edges. Moreover, going over the edges in the boundless simulation means that once the mobile node reaches the boundary from any side of the simulation area, it does not bounce the same as in the other models, but it disappears from the side and reappears from the other side continuing moving with the same direction, which makes the simulation area look more like a tube than a plane. Figure 1 shows one interpretation of what the simulation area might look like.

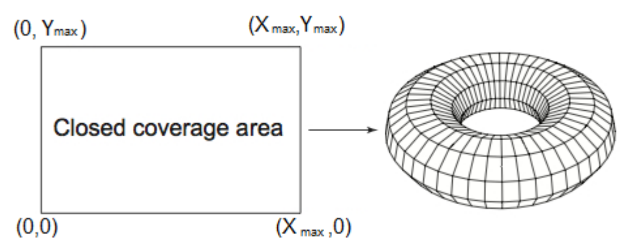

Figure 1. Simulation area of Boundless Simulation Area Mobility Model shown as a tube [18]

In our random generator, we used this method to avoid the border effect without affecting the randomness of node movement. Since the ns2 tool is based on TCL [26] object-oriented programming language, we decided to use Java programming language to build the new random generator. The new random generator generates a file readable by ns2 that has the definitions of all the movements of all the nodes during the simulation time. The Java code firstly generates the initial random positions of the nodes. The x,y

 ISBN: 978-1-61208-245-5

and z coordinates are given in ns2 TCL format, for example:

```
$node_(0) set X_ 323.81544267544473
$node_(0) set Y_ 576.9394231828528
$node_(0) set Z_ 0.000000000000
```

The movements' commands can be for either a node that does not cross the border(s) or a node that does cross the border. The nodes that do not go across the border(s) will have just one movement command as in the following example:

```
  $ns_ at 63.904477062 "$node_(0)
        setdest 118.025456608
        271.953011717 9.048003190"
```

On the other hand, the nodes that go across the border(s) will have three movement commands. The following is an example of the generated three commands by the custom random generator which represent the movement that has a jump from one border to the other through the movement.

```
  $ns_ at 102.779352050 "$node_(0)
setdest
        247.578795505 0.000000001
        12.273796854"
$ns_ at 113.299385328 "$node_(0) setpos
        247.578795505 999.999999999"
$ns_ at 113.299385328 "$node_(0) setdest
        247.984526589 998.357412892
        12.273796854"
```

The first line represents the movement from the current position to the border, the second line represents the jump from one border to the opposite border and the last line represents the movement from the border after the jumping to the destination.

The speed of movement is the same in the first and the third line which are making the movement to be exactly the same before the jump and after it. Figure 2 shows the three movements that happen once the node across the border(s) generated by the custom random generator. Algorithm 1 describes in details the algorithm used in the new custom movement generator.

We modified ns2 adding a new setpos command (analogous to setdest) that allows for placing a node at specified location during the simulation.

V. RESULTS

We investigated the randomness performance and the packet transmissions of both the setdest utility and the custom generator. The randomness performance of both generators is shown in Figure 3.

In this figure, the VMR values represent the randomness in the distribution of mobile nodes at different times in the

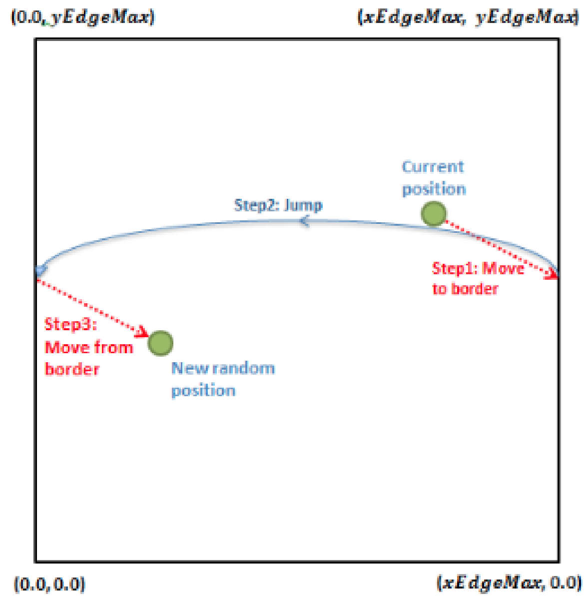

Figure 2. Three steps those nodes have to follow once they reach boundary

simulation. The results obtained with the custom generator are closer to the VMR value of one than the values obtained from using the setdest utility. Taking in to account the main deficiency of the setdest utility, which is the tendency of placing the mobile nodes in the center of the simulation area we also investigated the packet transmissions in a simulated mobile networks controlled by both generators in two ways: we measured transmitting the packets through the center and along the edges of the simulated area. Once the packets were transmitted through the center, there was no significant difference between the movement generators while using the 60 nodes, but with the 30 and 20 nodes there were significant differences with the setdest utility showing more packets delivered throught the center, suggesting some clustering of nodes in the center (Table I). Transmitting the packets along the edges (results are shown in Table II), we observed a significant difference between the two generators in the experiments with 30 and 60 nodes.

For example, when transmitting through the centre in the experiments with 30 nodes we observed on average 7453 packets delivered in the scenarios generated with setdest utility vs 5600 packets in the case when our custom generator was used. For the same scenarios with 30 nodes, transmissions along the edges gave 716 vs 2031 packets received. This shows a strong bias for packet delivery in the centre of the experimental area for the setdest utility: 7453 in the center vs 716 along the edges, almost 10 to 1 ratio. Similar comparison for the new generator gives only 2 to 1 ratio (5600 vs 2031), which is close to what would be expected based on the model described in [27].

Overall the simulation with the custom generator delivered

Algorithm 1 Pseudocode (Custom Generator)

```
(maxX, maxY) = simulation area dimensions
maxTime = max duration of node's movement
maxMovement = max distance of node's movement
currentTime = 0
(currentX, currentY) = random position

while Simulating do
    angle = rand(0..1) * 360°
    destX = rand(0..1) * cos(angle) * maxMovement //set the movement destination
    destY = rand(0..1) * sin(angle) * maxMovement
    MovementTime = rand(0..1) * maxTime
    speed = geometric distance (currentX, currentY) to (destX, destY)/movementTime

    if path does not cross the boundary then
        writeMovement ($ns_ at current Time "$nodeX setdest destX, destY, speed")
    else
        if path crosses the boundary then
            (interceptX, interceptY) = boundary intercept
            distToBound = geometric distance to the boundary in the direction of the movement path
            writeMovement ($ns_ at currentTime "$nodeX setdest interceptX, interceptY, speed")
            timeAtBoundary = currentTime + distToBound/speed
            writeMovement ($ns_ at timeAtBoundary "$nodeX setpos interceptX, interceptY")
            destX = destX % maxX
            destY = destX % maxY
            writeMovement ($ns_ at timeAtBoundary "$nodeX setdest destX, destY, speed")
            (currentX, currentY) = (destX, destY)
            currentTime = currentTime + movementTime
        end if
    end if
end while
```

packets better and more uniformly than the setdest utility.

Table I
AVERAGE PACKET DELIVERY FOR TRANSMITTING THROUGH THE CENTER (WITH 95% CONFIDENCE INTERVAL)

	60 Nodes	30 Nodes	20 Nodes
Setdest Utility	9595 (9459 - 9731)	7453 (6724 - 8182)	3878 (2945 - 4811)
Custom Generator	9490 (9371 - 9610)	5600 (5150 - 6050)	1886 (1325 - 2448)

Table II
AVERAGE PACKET DELIVERY FOR TRANSMITTING ALONG THE EDGES (WITH 95% CONFIDENCE INTERVAL)

	60 Nodes	30 Nodes	20 Nodes
Setdest Utility	4233 (3664 - 4802)	716 (478 - 955)	384 (193 - 576)
Custom Generator	8085 (7933 - 8238)	2031 (1712 - 2351)	332 (197 - 468)

VI. CONCLUSION

We investigated the performance of the popular setdest utility used in the ns2 network simulator. The movement generated with setdest utility tends to cluster the mobile nodes in the center of the experimental area. This has an effect on the VMR coefficient used to measure randomness of the positions of points in the simulation. We proposed and demonstrated the advantage of new random motion generator for use in ns2 simulator. Testing the new generator shows a marked advantage over the standard setdest utility and should improve the quality of MANET simulation models when randomness of node movement is required.

ACKNOWLEDGMENT

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

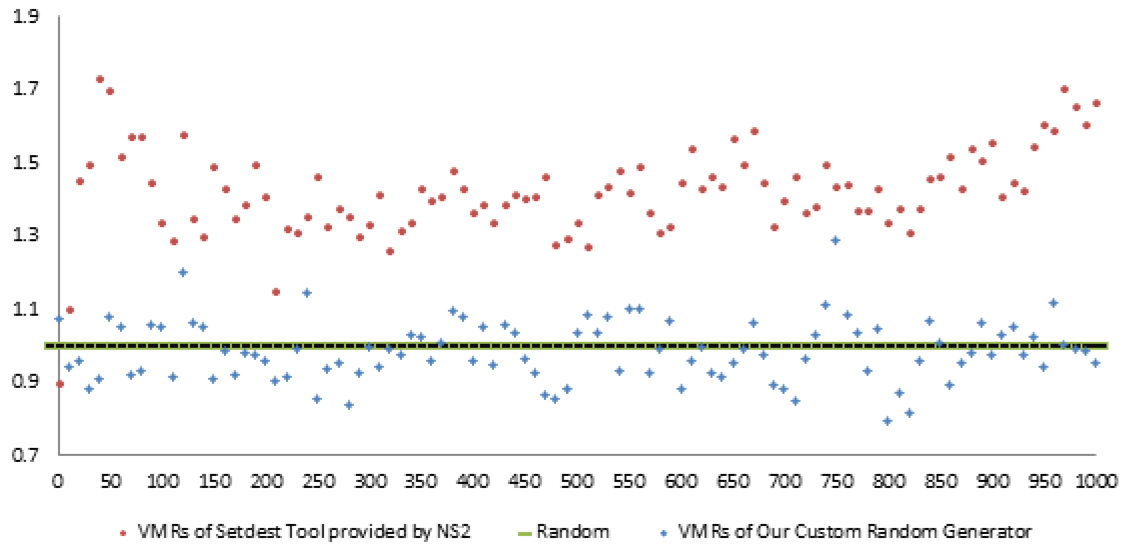

Figure 3. Comparison of the 101 means (of five VMR runs) of the setdest utility and the custom generator

REFERENCES

[1] Wikipedia, the free encyclopedia (January 2012), Mobile ad hoc network, http://en.wikipedia.org/wiki/Mobile_ad_hoc_networks (accessed November 2012)

[2] R. Suprio, Realistic mobility for MANET simulation, Masters thesis, The University of British Columbia, 2003.

[3] Wikipedia, the free encyclopedia (November 2009), Basic Station, http://en.wikipedia.org/wiki/Base_station (accessed October 2012)

[4] Z. J. Haas, "A new routing protocol for the reconfigurable wireless networks," in 1997 IEEE 6th International Conference on Universal Personal Communications Record, San Diego, CA, USA, 12-16 Oct. 1997, vol.2, pp. 562-566.

[5] Wikipedia, the free encyclopedia (November 2010), Wireless Ad-hoc Network, http://en.wikipedia.org/wiki/Wireless_ad-hoc_network (accessed December 2012)

[6] N. S. Dalton and R. C. Dalton, "The theory of natural movement and its application to the simulation of mobile ad hoc networks (MANET)", in Proc. the Fifth Annual Conference on Communication Networks and Services Research 2007, Fredericton, N.B., pp. 359-363.

[7] Y. Zhang and W. Li, "An integrated environment for testing mobile ad-hoc networks", in Proc. the Third ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc '02), Lausanne, Switzerland, June 2002, pp. 104-111.

[8] V. Davies, Evaluating Mobility Models Within an Ad Hoc Network, Masters thesis, Colorado School of Mines, Colorado, 2000.

[9] T. Camp, J. Boleng, and V. Davies, "A survey of mobility models for ad hoc network research", Wireless Communications & Mobile Computing, Special Issue on Mobile Ad Hoc Networking: Research, Trends and Applications, vol. 2, no. 5, pp. 483-502, 2002.

[10] B. J. Culpepper and H. C. Tseng, "Sinkhole intrusion indicators in DSR MANETs, in Proc. First International Conference on Broadband Networks 2004, San Jose, CA, USA, 25-29 October 2004, pp. 681-688.

[11] N. Frangiadakis, M. Kyriakakos, S. Hadjiefthymiades, and L. Merakos, "Realistic mobility pattern generator: design and application in path prediction algorithm evaluation," in Proc. The 13th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications 2002, Lisboa, Portugal, 15-18 Sept. 2002, vol.2, pp. 765-769.

[12] P. Caballero-Gil, C. Caballero-Gil, J. Molina-Gil, and C. Hernandez-Goya, "A simulation study of new security schemes in mobile Ad-hoc NETworks", in EUROCAST 2007, LNCS, vol. 4739, R. Moreno Daz, F. Pichler, A. Quesada Arencibia, Ed., Heidelberg: Springer, 2007, pp. 73-81.

[13] J. Hu and R. Marculescu, "DyAD: smart routing for networks-on-chip", in Proc. the 41st annual conference on Design automation, San Diego, CA, USA, June 07-11, 2004, pp. 260-263.

 ISBN: 978-1-61208-245-5

[14] A. L. Cavilla, MANET extensions to ns2. Available: http://www.cs.toronto.edu/andreslc/software/MANET extensions.tgz (accessed November 2012)

[15] B. Carbone, Routing Protocols for Interconnecting Cellular and Ad Hoc Networks, Masters thesis, Universite Libre De Bruxeles, Faculte des Sciences, Department d'Informatique, 2006.

[16] J. Illian, A. Penttinen, H. Stoyan, and D. Stoyan, Statistical analysis and modelling of spatial point patterns. West Sussex, England: John Wiley & Sons Ltd, 2011.

[17] R. R. Roy, Handbook of mobile ad hoc networks for mobility models (1st Ed.), New York, NY: Springer Science Business Media, 2011.

[18] J. Ariyakhajorn, P. Wannawilai, and C. Sathitwiriyawong, "A Comparative Study of Random Waypoint and Gauss-Markov Mobility Models in the Performance Evaluation of MANET," International Symposium on Communications and Information Technologies, 2006, ISCIT '06, Bangkok, Thailand, , Oct. 18 2006 - Sept. 20 2006, pp. 894-899.

[19] B. Liang and Z. J. Haas, "Predictive distance-based mobility management for PCS networks," Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies INFOCOM '99, New York, NY, USA, 21-25 Mar 1999, vol.3, pp. 1377-1384.

[20] C. Chiang, Wireless Network Multicasting. Ph.D. thesis, University of California, Los Angeles, 1998.

[21] F. Bai and A. Helmy, "A Survey of Mobility Models in Wireless Ad hoc Networks", in Wireless Ad Hoc and Sensor Networks, Chapter 1, Kluwer Academic Publishers, June 2004, pp. 1-29.

[22] M. Pascoe, J. Gomez, V. Rangel, and M. Lopez-Guerrero, "Route duration modeling for mobile ad-hoc networks", ACM Wireless Networks Journal (WiNet), 16(3), 2010, pp. 743-757.

[23] Wikipedia, the free encyclopedia (June 2012), Constant bitrate, http://en.wikipedia.org/wiki/Constant_bitrate (accessed December 2012)

[24] Wikipedia, the free encyclopedia (May 2012), User Datagram Protocol, http://en.wikipedia.org/wiki/User_Datagram_Protocol (accessed December 2012)

[25] Wikipedia, the free encyclopedia (December 2011), Ad hoc On-Demand Distance Vector Routing, http://en.wikipedia.org/wiki/Ad_hoc_On-Demand_Distance_Vector_Routing (accessed December 2012)

[26] Wikipedia, the free encyclopedia (December 2011), Tcl, http://en.wikipedia.org/wiki/Tcl (accessed October 2012)

[27] C. Bettstetter and J. Zangl, "How to achieve a connected ad hoc network with homogeneous range assignment: an analytical study with consideration of border effects", 4th International Workshop on Mobile and Wireless Communications Network, Stockholm, Sweden, Sept. 2002, pp. 125-129.

 ISBN: 978-1-61208-245-5