M2ANET Simulation in 3D in NS2

Nasir Mahmood, John DeDourek, Przemyslaw Pochec
Faculty of Computer Science
University of New Brunswick
Fredericton, Canada
e-mail: {nasir.mahmood, dedourek, pochec}@unb.ca

***Abstract*— In this paper, the enhancements to the ns2 source code adding a capability for modeling Mobile Ad Hoc Network (MANETs) and Mobile Medium Ad Hoc Network (M2ANETs) in three dimensions are described. ns2 is an open-source event-driven simulator designed specifically for research in computer communication networks. It allows modeling MANETs in two dimensions which imposes limits on investigating the MANET performance in the real world. Experiments were conducted using the modified ns2 simulator to determine the performance of M2ANETs in 3D, at different node densities and with different movement patterns. The results show that mobile nodes using 802.11 links and running DSR routing protocol can successfully operate as a mobile medium (M2ANET) in a 3D cube with dimensions less than 1500x1500x1500 meters. Simulation experiments also show that deploying of a 3D M2ANET in a multi-story building is not negatively affected by limiting the mobile node movement to horizontal planes corresponding to different floors in the building.**

Keywords-mobile network; simulation; NS2; 3D; MANET; M2ANET; Mobile Medium.

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].

A Mobile Medium Ad Hoc Network (M2ANET) is a particular configuration of a typical MANET proposed in [3], where mobile nodes are divided into two categories: (i) the forwarding only nodes forming the so-called Mobile Medium, and (ii) the communicating nodes, mobile or otherwise, that send data and use this Mobile Medium for communication. The advantage of this M2ANET model is that the performance of such a network is based on how well the Mobile Medium can carry the messages between the communicating nodes and not based on whether all mobile nodes form a fully connected network. An example of a M2ANET is a cloud of drones released over an area of interest facilitating communication in this area. Recently, a number of projects that match the M2ANET model have been announced; they include Google Loon stratospheric balloons [4] and Facebook high altitude solar powered planes [5] for providing Internet services to remote areas, and the Swarming Micro Air Vehicle Network (SMAVNET) project where remote controlled planes are used for creating an emergency network [6].

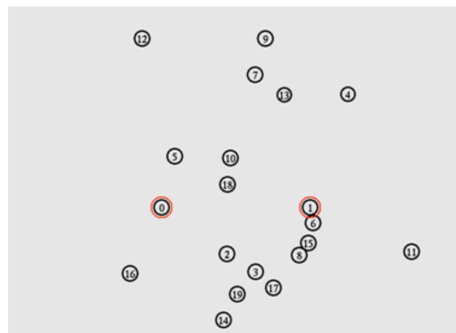

Figure 1. ns2 simulation screen of a MANET in 2D.

The existing simulation environments generally lack 3D capabilities. This paper describes the modifications to the ns2 simulator (Figure 1), version 2.35, adding a 3D mobile network modeling capability. The modified simulator is then used to investigate sample 3D M2ANET scenarios including a number of mobile nodes moving randomly in a 3D volume bounded by a cube and hypothetical scenario of mobile nodes moving on different floors in a multi-story building.

In Section II, the existing 3D network simulation tools are overviewed. Section III describes the modifications made to the ns2 simulator. Section IV describes a case study of wireless network simulation in 3D. Conclusions are presented in Section V.

II. STATE OF THE ART

A MANET is comprised of interconnected mobile nodes, which make use of wireless communication links for multi-hop transmission of data. The mobility plays a paramount role in the operation of a mobile network, however most scenarios referred to in the literature focus on 2D [7]. One of the few tools available for modeling node mobility in 3D is MobiSim [8]. It is a mobility management tool designed to produce movement trace files, Z coordinate included,

 ISBN: 978-1-61208-371-1

which can then be used for analysis or as an input to a simulator. As it generates only the movement traces, it cannot be used as a data network simulator. OPnet Modeler [9], in particular the Wireless Suite for Defense, includes integrated tools that incorporate the terrain modeling, node mobility and 3D visualization (3DNV), but, as a proprietary package, OPnet is expensive to license and lacks extensibility needed for experimentation with novel network mobility patterns [10]. ns2 is an open source simulator with built-in tools for modeling wired and wireless networks and for traffic visualization [11]. In its current version it does not support wireless node modeling in 3D.

In simulation, a random mobility is often used as a reference case scenario, mostly because of the relative ease of implementing it in a simulator. One of these popular models is the Random Way Point (RWP) model available in ns2 [10]. Nodes are moved in a piecewise linear fashion, with each linear segment pointing to a randomly selected destination and the node moving at a constant, but randomly selected speed.

III. MODIFICATIONS TO NS2

ns2 is an open source simulator consisting of over 300,000 lines of code in a number of libraries. The code uses a number of coding conventions including the representation of a mobile node's location. Mobile node coordinates are designated with variables X_, Y_ and sometimes Z_. As the simulator is designed for modeling 2D environments, if and when Z_ is used it is set zero.

The search of ns2 source files for variables X_, Y_ and Z_ followed by code inspection resulted in the following candidate files for modifications listed in Table I.

TABLE I: MODIFIED SOURCE CODE FILES

File	Methods
Mobilenode.cc	MobileNode( ) command( ) bound_position( ) set destination3d( ) update_position( ) log_movement( )
Mobilenode.h	getVelo( ) destZ( ) set_destination3d( ) initialized( )
Topography.cc	Load_cube( ) command( )
Topography.h	Load_cube( ) lowerZ( ) upperZ( ) Topography( )

In order to move a node in 3D space, a 3D topography needs to be defined. The maxZ of type double was defined and initialized to 0.0 in the Topography constructor where the maxX and the maxY are already initialized to 0.0. There are two getter methods defined for the maxZ in the topography header file. One is lowerZ, which always returns 0.0, and upperZ, which returns the product of maxZ and grid_resolution, where grid_resolution has default value of 1 unless initialized with different value using a TCL script.

A new method named load_cube was defined, which accepts the arguments x, y, z, and res, which are used to initialize maxX, maxY, maxZ, and grid_resolution, respectively. This load_cube method is called from the TCL script written for the purpose of running a 3D simulation experiment.

In the mobilenode header, X_ , Y_ , and Z_ are already defined in the original version of ns2 and they represent the position of a node in the topography, and the Z_ was neither implemented nor used. The dX_ , dY_ , and dZ_ are also already defined and represent a unit vector that specifies the direction of the mobile node movement. This vector is used to update the position of a node. The destX_ , destY_ , and destZ_ represent where the node is going. In order to get a value of the destZ_, a getter method was defined for the destZ_. The initialized method must check whether the topography and trace object are initialized and it also must check whether X_ , Y_ , Z_ are within the topography boundary. The Z_ condition was added in the initialized method to make sure the Z_ coordinate of the node is within the defined topography. In the getVelo method, the velocity factor for X_ and Y_ is already calculated using dX_*speed_ and dY_*speed_ respectively, so the velocity factor for Z_ which is the product of dZ_ and speed_ was added.

The new set_destination3d method is defined in mobilenode header, and takes X, Y, Z destination coordinates of a mobilenode along with the speed.

In the mobilenode class, the MobileNode constructor initialized destZ_ to 0.0. The constructor is also initializing variables like X_ , Y_ , Z_ , speed_ , dX_ ,dY_ , dZ_ , destX_ , and destY_ to 0.0 along with other variables. The bound_position method was also changed: it gets the lower and the upper bound of the X, Y, and Z coordinates of the topography and then checks whether the current position of the node is within the topography. If not the bound_position method will decrease or increase the coordinate value accordingly. In the bound_position method there was no handling of the Z coordinate, so this functionality was added to make it work for 3D. The set_destination3d method makes sure that all the class variables are initialized correctly, checks whether the new destination is within the topography boundary, and then makes a call to the update_position method, which calculates the current position of the node and then updates it to the new position. Then the update_position method makes a call to the log_movement method, which writes the new position in the log file. After the complete execution of the log_movement and update_position methods, the remaining set_destination3d method was executed which saves the value of the previous position of the node. The Z coordinate in the set_destination3d, update_position, and log_movement methods were implemented.

Modifications to the setdest movement file generation utility required modifications to the setdest class file.

The setdest class file does not have MAXZ defined so MAXZ is set to type double and initialized to 0.0 which is then used for the Z coordinate of the topography boundary. In order to set the value which is received as a command line argument for the Z coordinate of a topography another case was added in the switch statement for the Z coordinate that gets the command line argument value which comes after -z and saves it to MAXZ. The setdest utility terminates if the MAXZ value, which is received as a command line argument after –z, is equal to 0.0.

In setdest movement, the speed of the node is variable and does not exceed the max speed which is specified as a command line argument. We simplified the setdest utility and made the speed constant: the MINSPEED is set to MAXSPEED, and the speed is updated to be equal to MINSPEED in the RandomSpeed method. In modifying setdest we chose to keep the speed constant to make node movement modeling simpler and easier to trace and debug.

The RandomPosition method is called for node coordinate initialization. In this method, node X and Y coordinates are equal to uniform()*MAXX and uniform*MAXY respectively and the node Z coordinate is equal to 0.0. To make the RandomPostion method compatible with 3D Z is updated to uniform()*MAXZ.

The RandomDestination method is similarly updated. The RandomDestination method is used to assign a new destination to the node. In this method, node X and Y coordinates are equal to uniform()*MAXX and uniform()* MAXY and node Z coordinate is equal to 0.0 so it is updated to uniform()*MAXZ.

The print statement was updated so it also prints the z coordinate for node movement. Finally, the setdest3d method is used instead of the setdest method. The Z coordinate value of a node for Z coordinate initialization is already written to the output file.

The modifications to the ns2 source code and to the setdest utility were successfully tested with a number of validation tests which included: verifying if the 3D boundary limits are enforced, whether the nodes move correctly along paths defined in 3D, whether packets are received successfully in the 3D environment. As in standard ns2, at the end of a simulation run, the modified version generates a trace file which includes an additional field showing the Z coordinate on each node, like in the example below:

```
r -t 16.005981008 -Hs 0 -Hd 0 -Ni 0 -Nx
832.41 -Ny 842.41 -Nz 852.41 -Ne -
1.000000 -Nl AGT -Nw  -Ma 13a -Md 0 -Ms
1 -Mt 800 -Is 1.0 -Id 0.0 -It cbr -Il
500 -If 0 -Ii 18 -Iv 32 -Pn cbr -Pi 16 -
Pf 1 -Po 0
```

IV. EXPERIMENTS IN 3D

The enhanced version of ns2 described in this paper was used to model a mobile ad hoc network used as a mobile medium, i.e., M2ANET. Two groups of experiments were conducted: (i) free space experiments with one source and one destination node and the mobile medium nodes moving randomly inside a cube of different dimensions and (ii) multi-story building experiments with two sources and two destinations located at different floors of the building, and the mobile nodes moving either randomly or being limited to the planar motion of different floors.

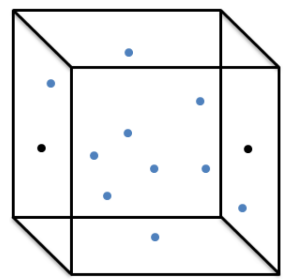

Figure 2. Free space movement in a cube.

The source and destination nodes were placed in the middle on the opposite sides of the cube (Figure 2) and 100 meters from the wall. In case of the two-story building scenario the locations were in the middle of the wall at each floor, with the source sending data to the destination on the same floor. For example, for a 2000x2000x2000 meter cube the source would be at (100, 1000, 1000) and the destination at (1900, 1000, 1000), see Fig. 1 for the general arrangement. In the experiments we used 500 byte packets sent every second (at Constant Bit Rate) over an 802.11 channel. The free space propagation model was used giving each node the communication range of 600 meters. The Dynamic Source Routing (DSR) protocol was used for routing.

Figure 3 shows the summary of simulation results for the first set of experiments in free space bounded by a cube of different dimensions: 1500, 2000, and 2500 meters. In the experiments the packet delivery was measured and averaged over 10 runs for each mobile node density. The first observation is that the performance of the network improves with the increasing number of mobile nodes in the mobile medium. For a 1500 m cube the highest packet delivery occurs at about 20 nodes, for a 2000 m cube at 40 nodes and for a 2500 m cube at 45 nodes. This is because for a larger cube more nodes are needed to build a path for sending packets from the source to the destination. The second observation is that the packet delivery declines when a large number of nodes are used. This is consistent with the DSR routing protocol performance where the route discovery overhead increases with longer paths and frequent disconnections in a larger network [12]. It is interesting to note that the rapid decline occurs at 75 nodes in a 1500 m cube, at 60 nodes in a 2000 m cube and at 50 nodes in a 2500 m cube.

.

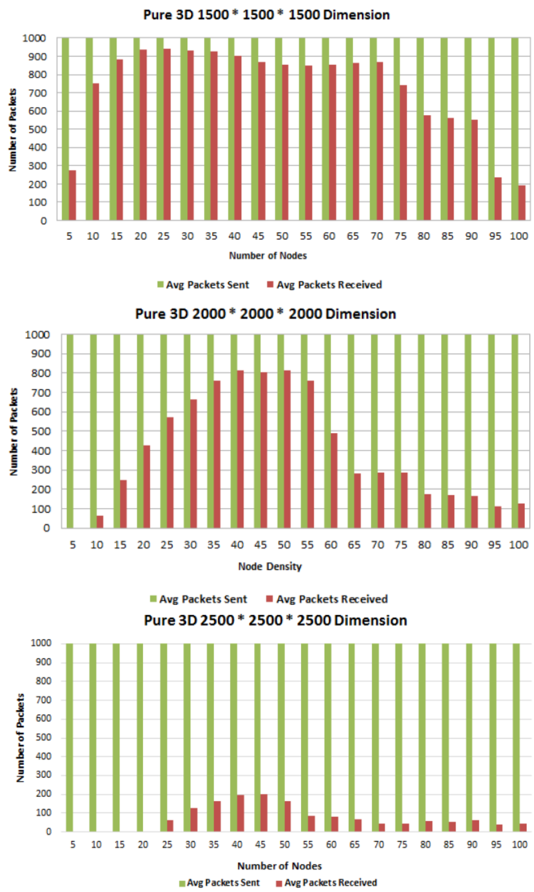

Figure 3. 3D experiments: free space movement in a cube

The two factors impacting the operation of the M2ANET in the experiments, namely improvement in performance due to more nodes available to create the network and decline in performance due routing overhead in a network with a large number of nodes, are opposing each other and have different impact depending on the size of the cube in which the network is deployed. For a smaller 1500 m cube the performance is stable for a range of node densities (20 to 70 nodes) while for a larger 2500 m cube the network never performs adequately because the delivery starts to decline due to DSR overhead even before the number of nodes would reach adequate density for forming a reliable path. This shows a limit on suitability of a DSR based network for creating a mobile medium covering a larger volume.

In the second set of experiments, we set out to demonstrate the use of the 3D network simulation tool in a more practical example of a multi-story building (Figure 4) where the movement of the nodes in not necessarily random in free space bounded by a cube but is limited to a planar movement on each floor, with the nodes on one floor interacting with the nodes on another floor when they are in range.

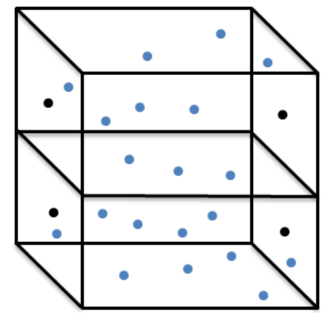

Figure 4. Multi-story building: two sources and two destinations with free space movement.

The experiments were run in a 1500x1500x1500 space with source and destination nodes on two planes: Z = 500 and Z = 1000. A total of 20 mobile nodes was used in each experiment. Three scenarios were tested: (i) random: all 20 nodes moving randomly in a 1500 m cube with no restrictions, (ii) independent: mobile nodes are divided in to two groups of 10 and each group moves randomly but is restricted to a different floor (plane), and (iii) synchronized: nodes are still divided into two groups, one group moves randomly on the upper floor and the other group on the lower floor shadows the movement of the first group with each node staying right below the node on the upper floor.

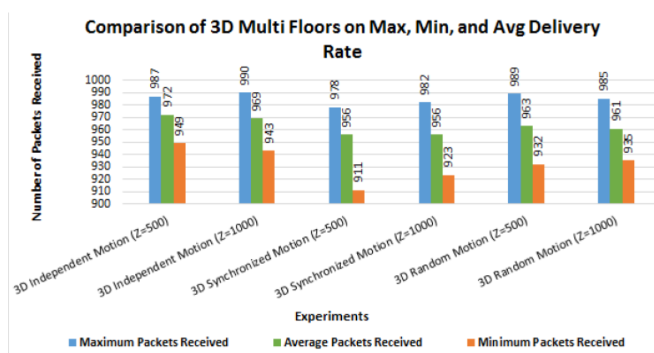

Figure 5. Comparison of delivery rate for three multi-story building scenarios.

The results of packet delivery averaged over 10 simulation runs (Figure 5) show a slight advantage of the scenario when the nodes are restricted to different floors and move independently. This can be explained by noting the fact that when totally random movement is used inside a cube, some nodes move into the corners of the cube which would limit their participation in forming a path form source to destination while restricting the node movements to each floor avoids this occurrence.

V. Conclusion and Future Work

In this paper, modifications to the ns2 version 2.35 simulator required for adding a new capability for modeling mobile networks in 3D were described. ns2 is an open source simulator and consists of over 300,000 lines of code. The modified ns2 has a capability to model MANET nodes and mobility patterns in 3D. A number of validation tests were performed to test the newly added 3D capability in the ns2 source simulator.

The modified simulator was used to experiment with a mobile medium network (M2ANET) in 3D. The tests showed that the mobile medium formed by mobile nodes running DSR over 802.11 connections with free space propagation range of 600 meters performs well in a 3D scenario covering a 1500x1500x1500 meter cube. In a larger cube, a high performance could not be achieved as a larger network with larger number of nodes required to fill in a larger space suffered from the performance decline due to an increasing overhead of the DSR routing protocol.

Experiments with modeling a multi-story building showed that the mobile medium performance would not decline when the mobile node movement is restricted to moving on each floor rather than randomly in free space. Based on our results, we suggest trying different existing ad hoc routing protocols in the modified simulator in 3D. It may be required to develop new routing schemes especially suited for forwarding in a mobile medium in 3D.

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] J. DeDourek and P. Pochec, "M2ANET: a Mobile Medium Ad Hoc Network", Wireless Sensor Networks: Theory and Practice, WSN 2011, Paris, France, Feb. 2011, pp. 1-4.
[4] S. Levy, "How Google Will Use High-Flying Balloons to Deliver Internet to the Hinterlands", Wired, June 2013, http://www.wired.com [retrieved: July 2014].
[5] J. Brustein, "Facebook's Flying Internet Service, Brought to You by Drones", Bloomberg Businessweek, March 4, 2014.
[6] A. Jimenez Pacheco et al., "Implementation of a Wireless Mesh Network of Ultra Light MAVs with Dynamic Routing", IEEE GLOBECOM 2012, 3rd International IEEE Workshop on Wireless Networking & Control for Unmanned Autonomous Vehicles 2012, Anaheim, California, USA, 2012, pp. 1591 - 1596.
[7] 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, v2, 2008, pp. 54-61
[8] S. M. Mousavi, H. R. Rabiee, M. Moshref, and A. Dabirmoghaddam, "MobiSim : A Framework for Simulation of Mobility Models in Mobile Ad-Hoc Networks", The 3rd IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (IEEE WiMob 2007), New York, USA, October 8-10, 2007, p. 82.
[9] OPnet Modeler, www.opnet.com [retrieved: June 2014]
[10] Ch. R. Rowell, "Modeling Computer Communication Networks in a Realistic 3D Environment", MCS Thesis, Department of Electrical and Computer Engineering, Airforce Institute of Technology, Wright-Patterson Air Forece Base, 2010, Ohio, USA.
[11] H. Ekram and T. Issariyakul, Introduction to Network Simulator NS2, Springer, 2009.
[12] A. M. Kanthe, D. Simunic, and R. Prasad, "Comparison of aodv and dsr on-demand routing protocols in mobile ad hoc networks", Emerging Technology Trends in Electronics, Communication and Networking ET2ECN 2012 1st International Conference on, Dec 2012, pp. 1-5.

 ISBN: 978-1-61208-371-1