

NetTopo

——Mobility Module

Lei Shu, Yuanbo Han

Documentation

Version: December 14, 2009

1 Contact Information

Email: lei.shu@ieee.org

yuanbo.han@gmail.com

URL: <http://www.semanticreality.org/nettopo/index.htm>

Postal address:

Digital Enterprise Research Institute,
National University of Ireland,
Galway, Ireland

2 Introduction

NetTopo is Java software, which is an open source research-oriented simulator & visualizer designed to test and validate algorithms for wireless sensor networks. The mobility module of NetTopo aims to let the sensor motes move within your distribution to simulate some kind of scenarios, thus helping to test, validate and optimize algorithms.

3 Parameter Specification (Mobility Module)

After you have successfully run the NetTopo application, there is a menu item named “Mobility”. The drop-down menu includes “RandomWaypoint” and “Stop”. Right parameters will help you to simulate the scenario right for you, and the “Stop” item will be usable after you have successfully run a RandomWaypoint scenario. The following will give you a further specification of the parameters after you select the item of “RandomWaypoint”.

1) nodesID

There are three input way. But you should pay attention to that you should not combine any of the following three ways together. You should use it respectively. Also, the node id you input should be integer and it should exist.

- You input the node separated by comma ids respectively. For instance, you want to let the node 5,15,25,52 to move, then you just input “5, 15, 25, 52”, the application will know what you have input.
- You input a range. For instance, you want to let the nodes between 20-56 move, you just input “20-56”.
- You can select the all checkbox to let all the nodes move.

2) duration

This parameter tells the application how long you are going to let the mobile nodes in the application move. The duration should be positive.

3) randomSeed

This helps the NetTopo to generate the random position of the nodes in the scenario. It should be non-negative integer or long.

4) ignorance

The ignorance is to help to generate the mobile position of the nodes in the scenario much better. And I advice you it should be larger than 6000. But if you input an integer less than it, it will be no problem.

5) attractorFields

The input numbers attractorFields parameter should be divided by 5. And every input should be separated by comma. This tells you that every 5-parameter represents an attractor. And they are <x-coordinate>, <y-coordinate>, <intensity>, <x standard deviation>, and <y standard deviation>. <x-coordinate> and <y-coordinate> represent the center position, the intensity represents that intensity of the attractor. If you have more than one attractor, the intensity will decide the probability of attracting the nodes. The bigger value, there will be more chance to attract nodes. <x standard deviation> and <y standard deviation> represent the deviation. Pay attention to that this should not be used together with distriectedArea.

6) distriectedAreas

There is no distriected Area by default. But if you want to implement a distriected area, there are two types existed: circle and rectangle. The parameters you input should be meet the requirement, and they are separated by comma. You should pay attention that the distriected area should not exist with the attractorFields. If the attractorFields exist, the distriectedArea should not work.

- 2 parameters. Represent the center node id and the half width of the distriected area. For instance, “4, 50” represents the coordinate of the node 4 is the center and the half width is 50 pixel.
- 3 parameters. Represent the center node id and the width, height of the distriected area. For instance, “4, 40, 50” represents the coordinate of the node 4 is the center, the half width is 40 pixel and the half height of the area is 50.
- 4 parameters. Represent the x and y coordinate of the center, then the width and height of the distriected area. For instance, “40, 50, 40, 50” represents the center’s coordinate is (40,50), the half width is 40 pixel and the half height of the area is 50.

7) minSpeed

Min speed is designed to describe the node min speed. It should be non-negative float or integer. If it is too small, maybe your mobile nodes will change a very short distance every second. But it should be smaller than maxSpeed.

8) maxSpeed

Max speed is designed to describe the node max speed. It should be non-negative float or integer. If it is too small, maybe your mobile nodes will change a very short distance every second. But it should be larger than minSpeed.

9) maxPause

Max pause describes the max pause after every direction-changed move. The node won't pause every time the nodes move. When the direction change, then maybe it will pause for less than maxPause time.

10) dimension

There are three dimension input, they are 1,2,3. You should put any one of them. But you should pay serious attention to this input. 3 is no problem for every scenario. For the nodes will move along any possible direction. But 1 or 2 is what you should take care. You should make sure that the node will move into the districted area if you input 1 or 2. If one node can't move into the districted area along x or y axis, the NetTopo will be in a loop to try to find a proper position but it won't find one. So please be aware of this. But 3 is no problem for any districted area.

- 1 stands for the node will move along the x-axis.
- 2 stands for the node will move along the x-axis, y-axis respectively. That means this time the node move along x-axis, but next time maybe it will move along the y-axis.
- 3 stands for the node will move randomly in the canvas. It will not move along x or y axis.

4 CKN Algorithm

After you have built a wsn and you implemented nodes, you can use the drop-down menu of Algorithm, select the menu item "CKN", there is the integration of CKN and TPGF.

- "CKN_TPGF_ConnectNeighbors": which you can use to see the sleep and awake nodes. The awake nodes will connect their awake neighbors.
- "CKN_TPGF_FindOnePath": if you have placed a source node and a sink node, this will run successfully.
- "CKN_TPGF_FindAllPaths": if you have placed a source node and a sink node, this will run successfully.

The parameter you may want to know that which can affect the output of the CKN. It is the k which is initialized in the file named "CKN_MAIN.java". If you change the k value in that constructor, you will get different output.