5/3/2023

An Analysis of the Hot Spot Mobility Model

Nathan Wang, Shan Jiang

Table of Contents

Introduction

In this paper, an analysis of the newly implemented Hot Spot Mobility Model is performed. The functions behind the classes are discussed in detail and some data analysis is performed to showcase the implementation of the model. Finally, some considerations into how to improve the overall system in the future are analyzed.

Creating the Hot Spot Mobility

The idea behind the hot spot mobility model was to simulate how certain locations in real life were more likely to be traversed to as destination points. The new class contains a constructor, initializing functions, a function for traversal, a function to create the hot spots, and a few getter and setter functions.

Hot Spot Constructor

For the constructor of the new hot spot mobility model, only 4 attributes were required. The first attribute that was required was speed/velocity. The second attribute that was required was the pause constraint, which was used to have the nodes pause for a certain time once the waypoint or destination hot spot was reached. The third attribute required is the string representation of the hotspot locations. This string would then be parsed by a function to create vector of hot spot locations. The final attribute required is the waypoints constraint, which determined how many waypoints are needed to be traversed prior to reaching the destination hot spot location.

Hot Spot Vector Creation Function

The createHotspots function was designed to help parse the string being passed in from the constructor. A substring loop is used to loop through the string, splitting the string into their x,y,z locations. A ns3 vector is then created holding these values and is added to the class's private Vector of ns3 vectors.

Initialize Functions

Like the other mobility models, the hot spot mobility model has two initialize functions. The first function is the DoInitialize function, which is only called once after instantiation. In this function, the node being initialized is set to a random hot spot location. The destination hot spot location that the node is designated to traverse to is also set; this location is purposefully set so that the starting location and the destination location will never be the same with a while loop checker. After the initialization of locations, the class's other private initialize function, DoInitializePrivate, is called along with the parent class Mobility Model's DoInitialize function. In the DoInitializePrivate function, the node is paused and scheduled to start traversal. After scheduling, the NotifyCourseChange function is called, and the node is unpaused.

BeginWalk Functions

The BeginWalk function is the update function for traversing the node. However, unlike most mobility models which traverse to one destination, the hot spot model is a bit more complex. Since the hotspot model requires the node to traverse to waypoints before reaching the destination hot spot, conditionals were implemented to determine whether to generate waypoints or traverse to the destination hot spot. If the number of waypoints traversed is less than the number specified, then a random location was generated between the current location and the midpoint between the current node position and destination location. After creating the new waypoint location, the counter for waypoints reached would be incremented. If the number of waypoints reached has been met, the node's next location would be set to the destination hot spot location. Afterwards, a new destination hot spot location that is not the same as the current destination hot spot location will be randomly assigned. After waypoint location assignment, the velocity is then calculated, and a new event is then scheduled for the time needed to traverse to the new location and the DoInitializePrivate function is called again. The function NotifyCourseChange is also called.

Data Analysis

For this project, the team decided to try simulating hot spot mobility in real life scenarios. In real life situations, hot spot mobility could model a person traversing from one location to another. In particular, the team wanted to analyze data rate loss as nodes traversed in hot spot mobility to simulate signal loss as people moved from signal "hotspots" to other hotspots. In the analysis, the team focused on a centralized AP node with one traversing node along with multiple AP nodes and one traversing node. 1 Traversing node was used for more efficient data collection although the model should be able to support multiple traversing nodes in theory.

1 AP Node and 1 Traversing Node

The idea behind the 1 AP node and 1 traversing node configuration was based off real-world situations where there would be one source point sharing data within a certain region. One such example would be like a single Wi-Fi router in a house. The team wanted to see the patterns in Minstrel Rate as the node traversed to randomly generated hotspots with the AP node at the center of the region. The hotspots were generated for at varying distances within 80 to 180 sized bounded rectangles. For data analytics, a scheduled event would occur every timestep which would calculate the distance from the AP node and the Minstrel rate would be analyzed.

```
nathanwang@lawn-143-215-62-226 ns-3.37 % ./ns3 run "scratch/OneAPTest.cc"
Initialized Location at 119:120:0
Simulation Time: 1.5    Station Location: (119, 120, 0) Distance: 169.027       Final Link Rate: 13 Mbps
Waypoint at 94.2573:134.644:0
Simulation Time: 2.5    Station Location: (116.672, 121.378, 0) Distance: 168.386       Final Link Rate: 26 Mbps
Simulation Time: 3.5    Station Location: (112.015, 124.134, 0) Distance: 167.229       Final Link Rate: 26 Mbps
Simulation Time: 4.5    Station Location: (107.359, 126.89, 0)  Distance: 166.24        Final Link Rate: 26 Mbps
Simulation Time: 5.5    Station Location: (102.702, 129.646, 0) Distance: 165.423       Final Link Rate: 26 Mbps
Simulation Time: 6.5    Station Location: (98.0454, 132.402, 0) Distance: 164.779       Final Link Rate: 26 Mbps
Simulation Time: 7.5    Station Location: (93.3888, 135.158, 0) Distance: 164.311       Final Link Rate: 26 Mbps
Simulation Time: 8.5    Station Location: (88.7322, 137.914, 0) Distance: 164.02        Final Link Rate: 26 Mbps
Waypoint at 60.8086:154.44:0
Simulation Time: 9.5    Station Location: (84.1326, 140.636, 0) Distance: 163.908       Final Link Rate: 26 Mbps
Simulation Time: 10.5   Station Location: (79.7815, 143.211, 0) Distance: 163.962       Final Link Rate: 26 Mbps
Simulation Time: 11.5   Station Location: (75.4305, 145.786, 0) Distance: 164.172       Final Link Rate: 26 Mbps
Simulation Time: 12.5   Station Location: (71.0795, 148.361, 0) Distance: 164.537       Final Link Rate: 26 Mbps
Simulation Time: 13.5   Station Location: (66.7284, 150.936, 0) Distance: 165.056       Final Link Rate: 13 Mbps
Simulation Time: 14.5   Station Location: (62.3774, 153.511, 0) Distance: 165.728       Final Link Rate: 6.5 Mbps
Simulation Time: 15.5   Station Location: (58.0263, 156.086, 0) Distance: 166.55        Final Link Rate: 6.5 Mbps
Simulation Time: 16.5   Station Location: (53.6753, 158.662, 0) Distance: 167.522       Final Link Rate: 6.5 Mbps
Destination at 21:178:0
Simulation Time: 17.5   Station Location: (49.6162, 161.064, 0) Distance: 168.56        Final Link Rate: 6.5 Mbps
Simulation Time: 18.5   Station Location: (45.7218, 163.369, 0) Distance: 169.673       Final Link Rate: 6.5 Mbps
Simulation Time: 19.5   Station Location: (41.8274, 165.674, 0) Distance: 170.898       Final Link Rate: 6.5 Mbps
Simulation Time: 20.5   Station Location: (37.9329, 167.978, 0) Distance: 172.234       Final Link Rate: 6.5 Mbps
Simulation Time: 21.5   Station Location: (34.0385, 170.283, 0) Distance: 173.678       Final Link Rate: 6.5 Mbps
Simulation Time: 22.5   Station Location: (30.1441, 172.588, 0) Distance: 175.227       Final Link Rate: 6.5 Mbps
Simulation Time: 23.5   Station Location: (26.2497, 174.893, 0) Distance: 176.877       Final Link Rate: 6.5 Mbps
Simulation Time: 24.5   Station Location: (22.3552, 177.198, 0) Distance: 178.628       Final Link Rate: 6.5 Mbps
Simulation Time: 25.5   Station Location: (18.4608, 179.503, 0) Distance: 180.475       Final Link Rate: 6.5 Mbps
Simulation Time: 26.5   Station Location: (14.5664, 181.808, 0) Distance: 182.415       Final Link Rate: 6.5 Mbps
Waypoint at 13.9824:177.204:0
Simulation Time: 27.5   Station Location: (13.6054, 179.847, 0) Distance: 180.385       Final Link Rate: 6.5 Mbps
Simulation Time: 28.5   Station Location: (14.21, 175.609, 0)   Distance: 176.208       Final Link Rate: 6.5 Mbps
Simulation Time: 29.5   Station Location: (14.8146, 171.371, 0) Distance: 172.036       Final Link Rate: 6.5 Mbps
Waypoint at 16.6595:158.441:0
Simulation Time: 30.5   Station Location: (15.5045, 166.536, 0) Distance: 167.283       Final Link Rate: 6.5 Mbps
Simulation Time: 31.5   Station Location: (16.3357, 160.71, 0)  Distance: 161.566       Final Link Rate: 6.5 Mbps
Simulation Time: 32.5   Station Location: (17.1669, 154.884, 0) Distance: 155.861       Final Link Rate: 6.5 Mbps
Simulation Time: 33.5   Station Location: (17.9981, 149.058, 0) Distance: 150.171       Final Link Rate: 6.5 Mbps
Destination at 23:114:0
Simulation Time: 34.5   Station Location: (18.7869, 143.529, 0) Distance: 144.785       Final Link Rate: 26 Mbps
Simulation Time: 35.5   Station Location: (19.5487, 138.19, 0)  Distance: 139.598       Final Link Rate: 26 Mbps
Simulation Time: 36.5   Station Location: (20.3105, 132.85, 0)  Distance: 134.427       Final Link Rate: 26 Mbps
Simulation Time: 37.5   Station Location: (21.0724, 127.511, 0) Distance: 129.275       Final Link Rate: 26 Mbps
Simulation Time: 38.5   Station Location: (21.8342, 122.171, 0) Distance: 124.143       Final Link Rate: 26 Mbps
Simulation Time: 39.5   Station Location: (22.596, 116.832, 0)  Distance: 119.034       Final Link Rate: 26 Mbps
Simulation Time: 40.5   Station Location: (23.3578, 111.492, 0) Distance: 113.952       Final Link Rate: 26 Mbps
Simulation Time: 41.5   Station Location: (24.1197, 106.152, 0) Distance: 108.899       Final Link Rate: 52 Mbps
Waypoint at 23.6225:122.419:0
Simulation Time: 42.5   Station Location: (24.4069, 105.795, 0) Distance: 108.615       Final Link Rate: 52 Mbps
Simulation Time: 43.5   Station Location: (24.1584, 111.062, 0) Distance: 113.699       Final Link Rate: 52 Mbps
Simulation Time: 44.5   Station Location: (23.9099, 116.329, 0) Distance: 118.798       Final Link Rate: 52 Mbps
Simulation Time: 45.5   Station Location: (23.6614, 121.596, 0) Distance: 123.913       Final Link Rate: 39 Mbps
Simulation Time: 46.5   Station Location: (23.4128, 126.863, 0) Distance: 129.04        Final Link Rate: 39 Mbps
Simulation Time: 47.5   Station Location: (23.1643, 132.13, 0)  Distance: 134.179       Final Link Rate: 39 Mbps
Waypoint at 22.2427:151.663:0
Simulation Time: 48.5   Station Location: (22.8937, 137.866, 0) Distance: 139.786       Final Link Rate: 39 Mbps
Simulation Time: 49.5   Station Location: (22.619, 143.688, 0)  Distance: 145.488       Final Link Rate: 39 Mbps
```

Figure 1: The Data Rate Analysis with Hot Spot configuration of 2 Waypoints and a bounded region of 180x180

After analyzing different size configurations for generating hot spots, the team noticed that as the size of the bounded rectangle continued to increase, the chances of data rate decreasing increased. This was because the node traversing to waypoints and destinations closer to the edge of the rectangle led to greater distance from the AP node, which caused drops in data rate. Another thing that the team noticed was that the number of hotspots did not necessarily improve upon data rate; this is again because hot spot locations could still be randomly generated to the edges of the rectangle. However, more nodes increased the chances that some hot spots could be generated closer to the AP node, which would increase the data rate.

Multiple AP Nodes and Traversing Nodes

The idea behind testing multiple AP nodes was to simulate cases where there would be multiple source-emitting nodes. One such example would be traversal between different buildings, each containing their own Wi-Fi routers.

Multiple AP Nodes Mobility Pattern Analysis

In this experiment, a single traversing node was set to wander between three AP nodes. An example trace output is shown in Appendix A. The green nodes depict the location of the AP nodes, and the red node marks the location of the traversing node. When the duration time is relatively small, the traversing node travels along a straight line between AP nodes (Appendix B). However, when the duration time increases, the traversing node's trajectory gradually spreads out. This could be caused due to the provided ns3 CalculateDistance function generating round-off error over time.

The next experiment that the team performed was regarding distance. For this experiment, distances were set to 10m, 50m, and 100m with a duration time of 20s. The three resulting maps were generated (Appendix C,D,E). As shown in Appendix C, a smaller distance results in a more concentrated distribution of AP and Traversing Node, and vice versa. When comparing overall concentration between hotspot and random waypoint, hotspot usually exhibited more dispersed traversing nodes compared to the latter. This effect becomes more noticeable as the APs get more concentrated.

The Original Method of Calculating the Average Throughput

After completing the basic analysis of the distribution of Multiple AP Nodes and Multiple Traversing nodes, a further analysis was performed the overall performance of the network. Specifically, the team wanted to observe the change in Throughput by changing the number of AP nodes and distance in the network. The procedure to calculate the Throughput is as follows：

1. Set a long enough duration time (default 20s) for the distribution to stabilize and extract the position and distribution of AP nodes and Traversing Nodes.
2. Connect each Traversing Node to its nearest AP Node.
3. Run the simulation and calculate the average Throughput of each downlink.

When running the experiment, the team noticed that there were some flaws with the initial procedures. In the experiments, the method often outputted extremely high or low throughput values, making the results ineffective in representing network performance. This result is hypothesized to occur because the throughput of downlink with only one single device tends to fluctuate significantly when the channel SNR reaches a threshold. Another problem with this approach is that in the real-world scenarios, multiple devices often simultaneously connect to one single AP node, which can throttle the bandwidth and power of the AP node. Therefore, the method mentioned is unable to effectively reflect the status of throughput fairness.

The Optimized Method of Calculating the Average Throughput

In response to the flaws in the previous method, the team designed and implemented the following method to optimize the calculation of the average throughput of the network:

1. Set the duration time (default 20s) for the distribution to stabilize and extract the position distribution of AP nodes and traversing nodes at this time.
2. Create a set of topologies for all AP nodes and connect all traversing nodes to the AP node in the topology with a downlink.
3. Run the simulation again to calculate the throughput of each downlink in real-time and record it when the network stabilizes.

4. Compare the throughput of each downlink, find the AP node with the best throughput for each traversing node, and connect them.

5. Perform a final simulation and calculate the average value for the throughput of all downlinks.

Appendix F shows an example topology of the throughput calculation and Appendix G shows the output of the program.

Results

After compiling the experimental data, the team plotted a graph of analyzing throughput with different constraints. When the number of AP nodes in the hotspot model was increased with other constraints bound, the average throughput values of the nodes decreased (Appendix H). This is most likely caused by a more equal distribution of throughput between other AP nodes. Based off Appendix I, when increasing the distance of the hotspot model with other constraints bound, the average throughput value of the nodes was observed to have decreased. This result is similar to the results found in 1 AP Node and 1 Traversing Node, since adding distance increases that chance that data rate will drop would also result in a drop in throughput. After comparing both graphs, it was noticed that the number of AP nodes affected the throughput more than changing the distance.

Finally, the team created a graph regarding throughput and AP node density (Appendix J). As the density of the AP nodes increases, the throughput also increases. However, after the density reaches a certain amount, the throughput plateaus.

Future Work

In the future, creating a more flushed-out and complete version of the hot spot mobility model which includes aspects as 3D traversal and non-straight-line traversal may help the model simulate real-world conditions better. The implementation of 3D traversal may help determine signal strength in mountainous/hilly regions or even strength in buildings. Implementing a non-straight-line traversal will allow for better simulation of waypoints such as traffic lights, stop signs, and other stops.

References

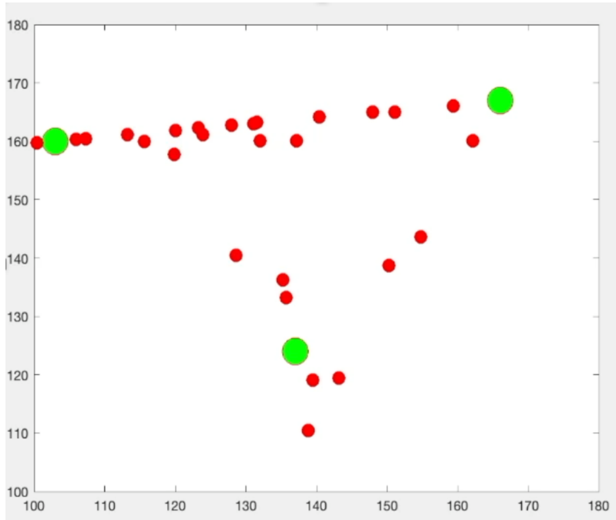
Hasib A, Fapojuwo A O. Mobility model for heterogeneous wireless networks and its application in common radio resource management[J]. IET communications, 2008, 2(9): 1186-1195.

Hyytia E, Lassila P, Virtamo J. A markovian waypoint mobility model with application to hotspot modeling[C]//2006 IEEE International Conference on Communications. IEEE, 2006, 3: 979-986.

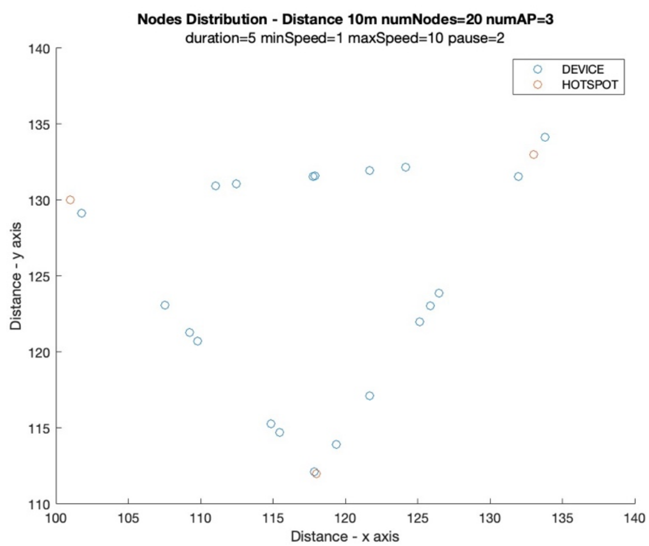
Kim, M., Kotz, D., & Kim, S. (2006). Extracting a Mobility Model from Real User Traces. In Proceedings of the 25th IEEE International Conference on Computer Communications (INFOCOM) (pp. 1–12). IEEE.

Morlot F, Elayoubi S E, Baccelli F. An interaction-based mobility model for dynamic hot spot analysis[C]//2010 Proceedings IEEE INFOCOM. IEEE, 2010: 1-9.

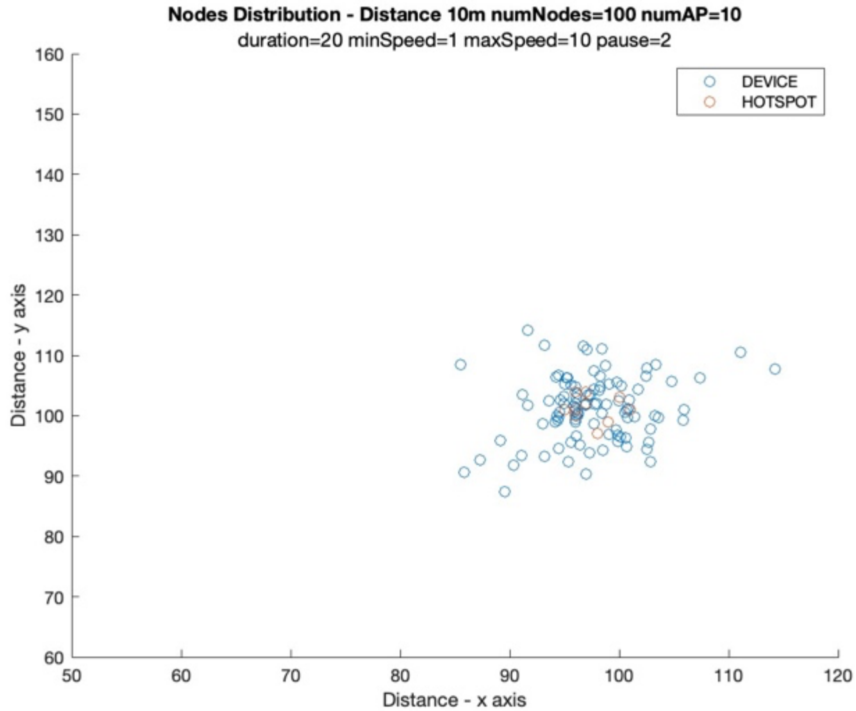
Appendix A: Single Traversing Node with 3 AP Nodes



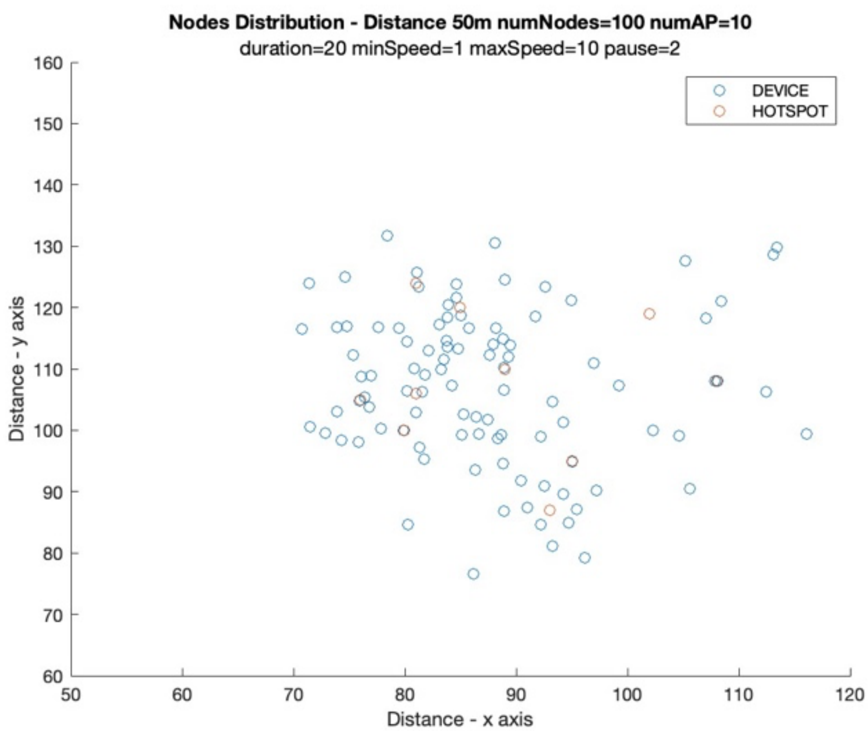
Appendix B: Node Distribution with Time of 20s



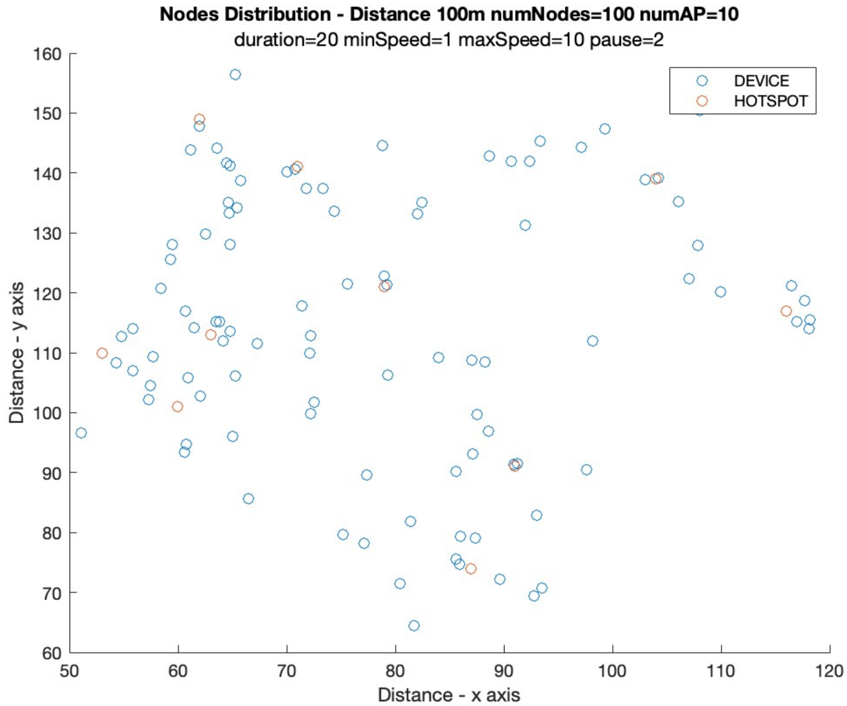
Appendix C: Node Distribution with Distance of 10



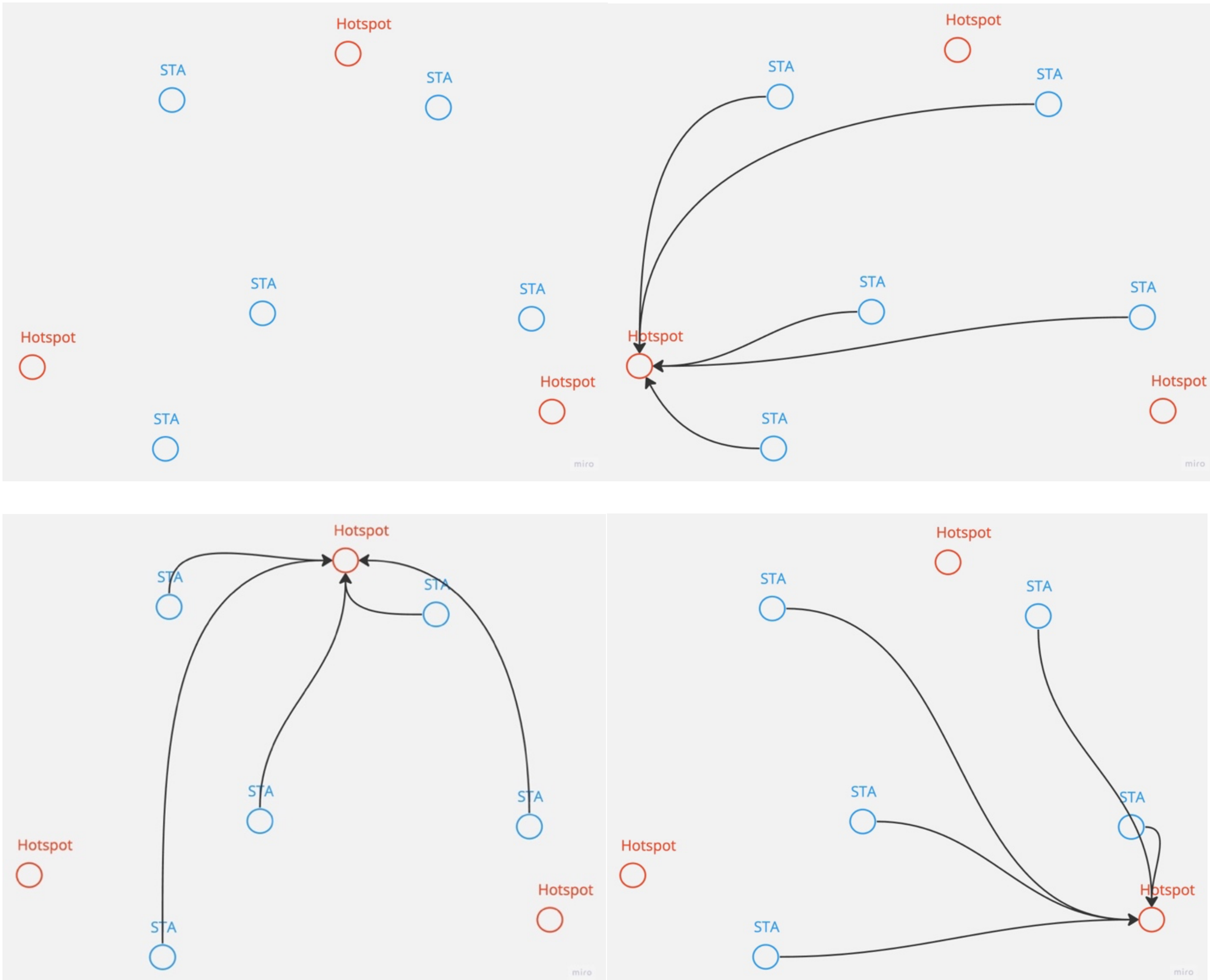
Appendix D: Node Distribution with Distance of 50



Appendix E: Node Distribution with Distance of 100



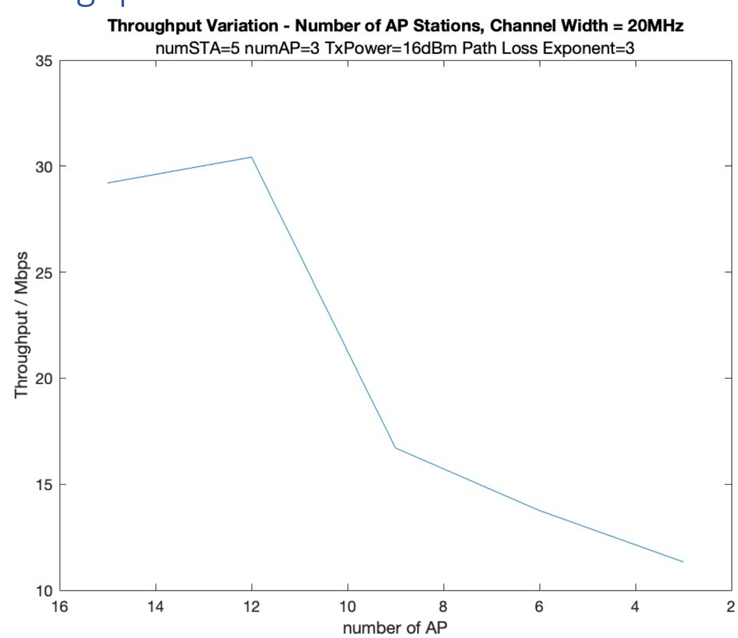
Appendix F: Example Topology for Average Throughput Computation



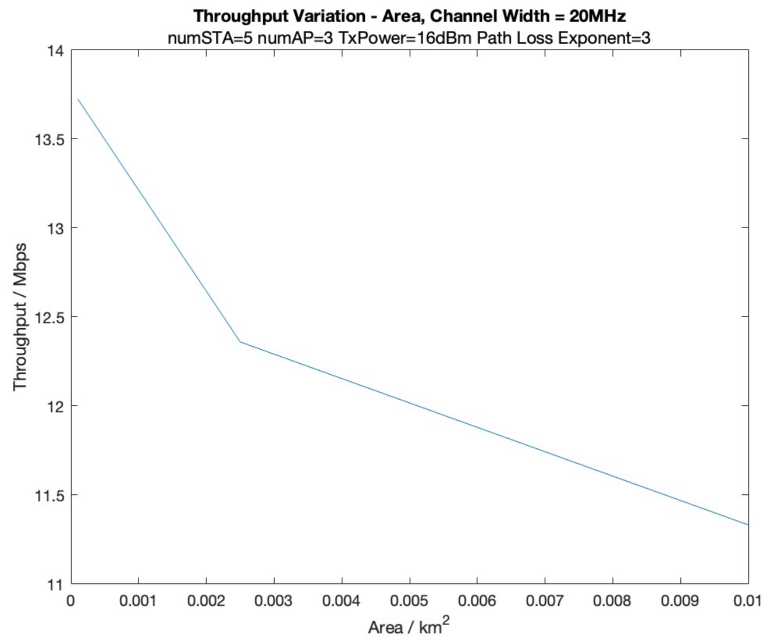
Appendix G: Command Line Output of Average Throughput Calculation

```
root@119afa9a93a8:/home/shan/ns-3-allinone/ns-3-dev# ./distance_test.sh
ap1
AP node starts at: 103:102:0
time, x, y, dataRate, Thoughput, NodeID
1.5,101,106,78,22.4461,1
1.5,109,103,78,0,2
1.5,107,97,78,0,3
1.5,105,111,78,20.0622,4
1.5,111,112,78,17.793,5
2.5,101,106,78,17.2278,1
2.5,109,103,78,14.0902,2
2.5,107,97,78,0,3
2.5,105,111,78,16.6052,4
2.5,111,112,78,17.0066,5
3.5,101,106,65,13.7708,1
3.5,109,103,65,13.8609,2
3.5,107,97,65,11.5999,3
3.5,105,111,65,13.7626,4
3.5,111,112,65,13.4349,5
4.5,101,106,78,13.3693,1
4.5,109,103,78,13.7626,2
4.5,107,97,78,13.7626,3
4.5,105,111,78,13.6724,4
4.5,111,112,78,13.9592,5
ap2
AP node starts at: 106:106:0
time, x, y, dataRate, Thoughput, NodeID
1.5,101,106,78,22.4461,1
1.5,109,103,78,0,2
1.5,107,97,78,0,3
1.5,105,111,78,20.0622,4
1.5,111,112,78,17.793,5
```

Appendix H: Throughput vs Number of AP Nodes



Appendix I: Throughput vs Distance



Appendix J: Throughput vs Density of AP Nodes