# Analysis of Random Geometric Graphs

Name: Da Cheng
Email: dcheng69@uwo.ca
ID: 251350918

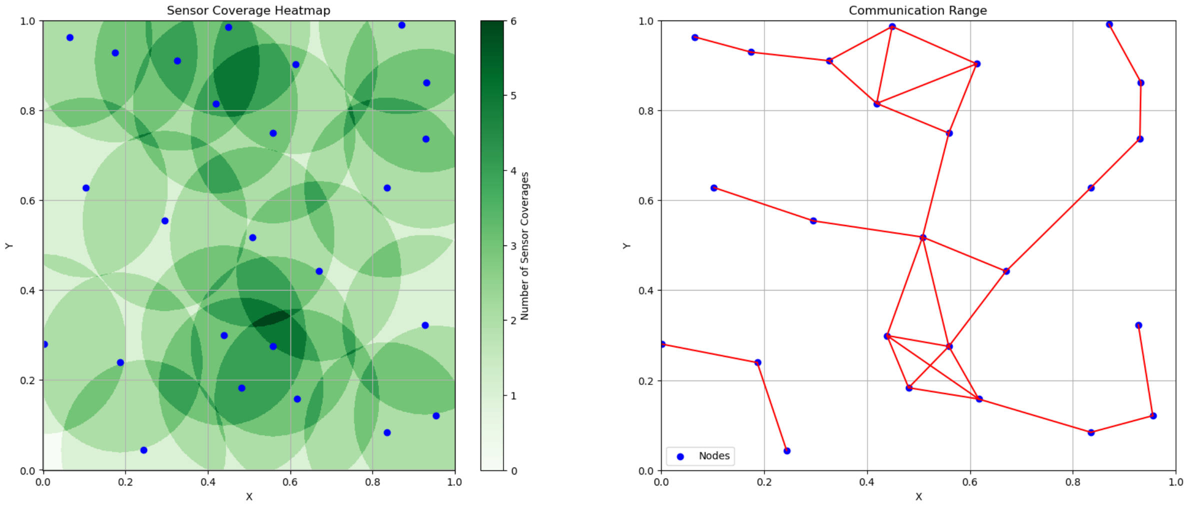
## Objectives

Evaluating the Performance of a pseudo-randomly generated graph， as an abstraction of a wireless sensor network. The nodes information are listed in the `nodes.txt` file. The Sensor Coverage and Communication Range plots are described below:

One thing to notice, the 25 nodes given are not fully connected!

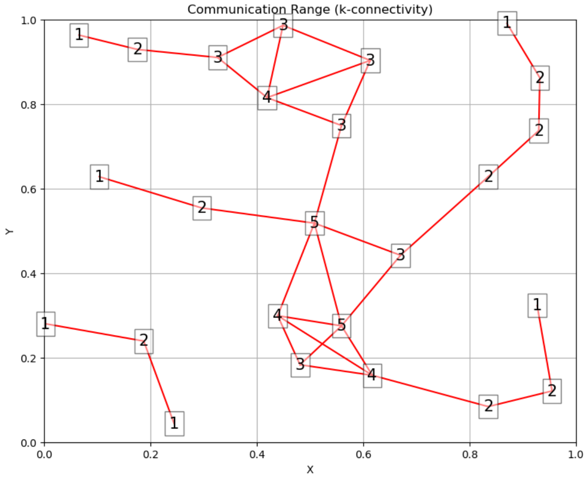
```
# nodes
0.439, 0.299
0.187, 0.239
# Communication Range
√((0.299−0.239)^2 + (0.439−0.187)^2) = 0.259L
```

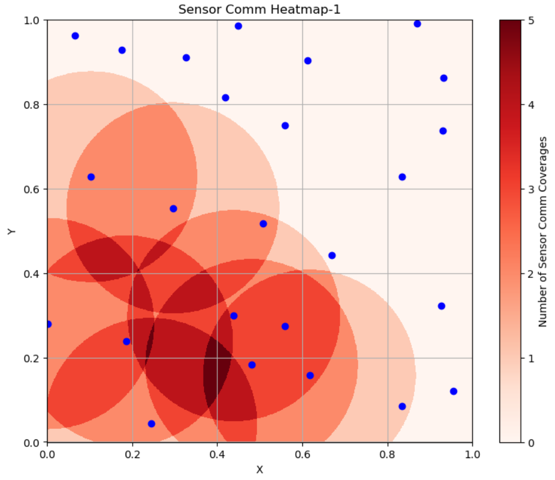
So in order to make the graph fully connected, one node need to added to connect two sub-nets.

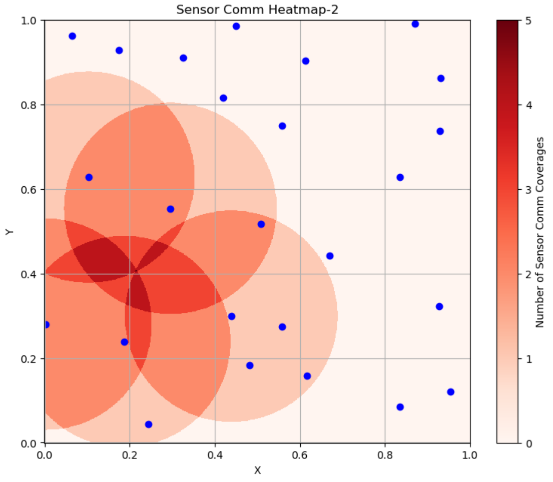


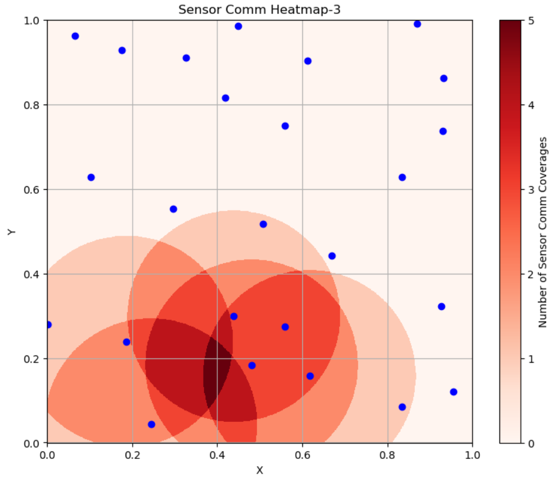
## Connectivity of the Network

As mentioned above, the connective of the original network is not **fully-connected**!. The `k-connectivity` of the original network is `1` which is shown in the **Communication Range** subplot, according to the **`Sensor Comm Heatmap-1`**, there are a few possible locations to put the new point, which are coloured in dark red. Further more, **`Sensor Comm Heatmap-2`** and **`Sensor Comm Heatmap-3`** shows two direction of putting the extra point.





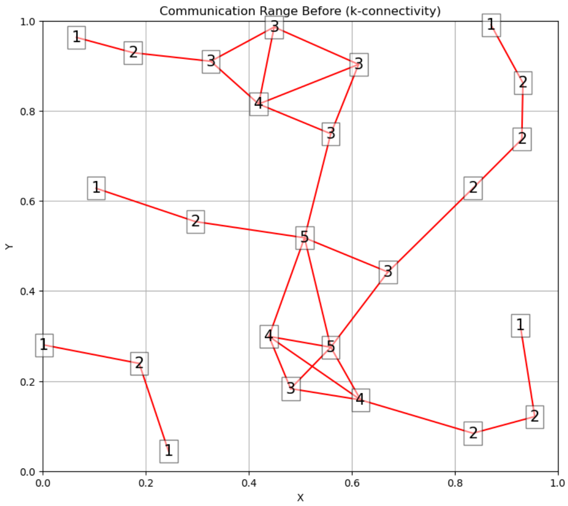


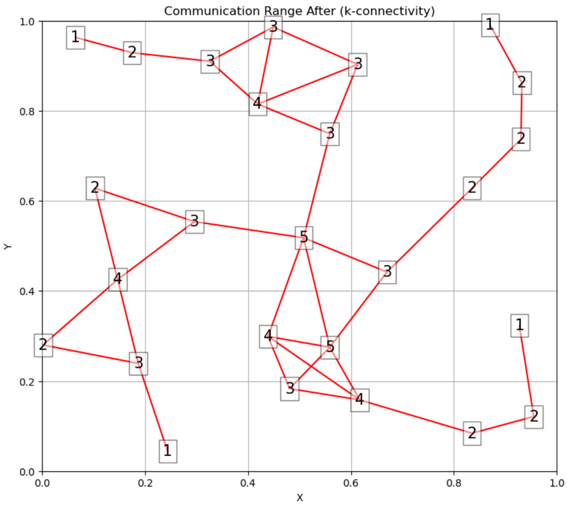


Combined with the `Sensor Coverage Hetmap` above, I decide to add one point that has the equal distance to the following four points:

```
# put one point that has equal distances to the following four points
(0.187, 0.239), (0.002, 0.28), (0.103, 0.628), (0.296, 0.554)

#For the x-coordinate:
(0.187 + 0.002 + 0.103 + 0.296) / 4 = 0.147

#For the y-coordinate:
(0.239 + 0.28 + 0.628 + 0.554) / 4 = 0.42525

#So, the centroid is approximately
(0.147, 0.42525)
```

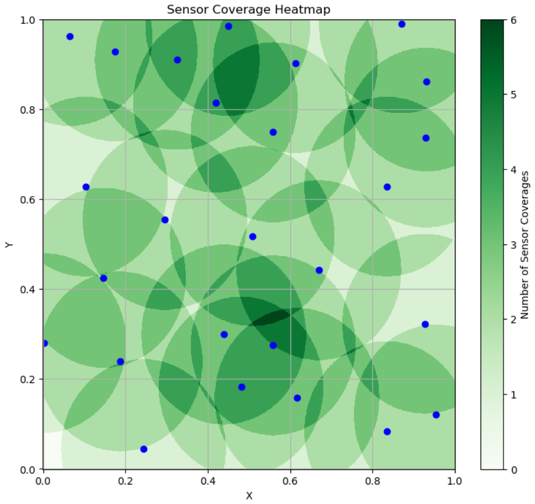
After Adding this point, the `k-connectivity` is still `1`, but the whole network becomes **fully-connected**!

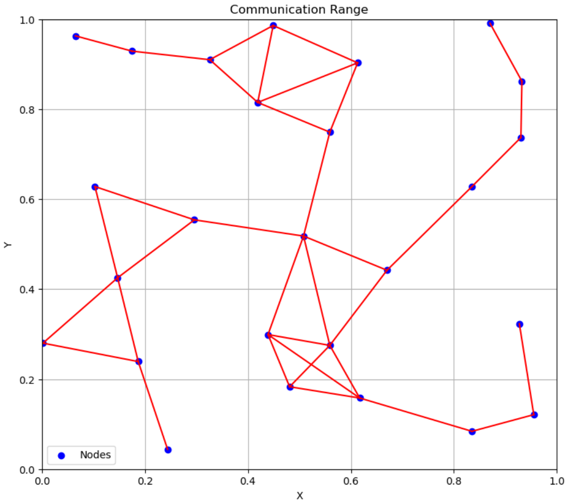




# Accurate Diagram of the Network

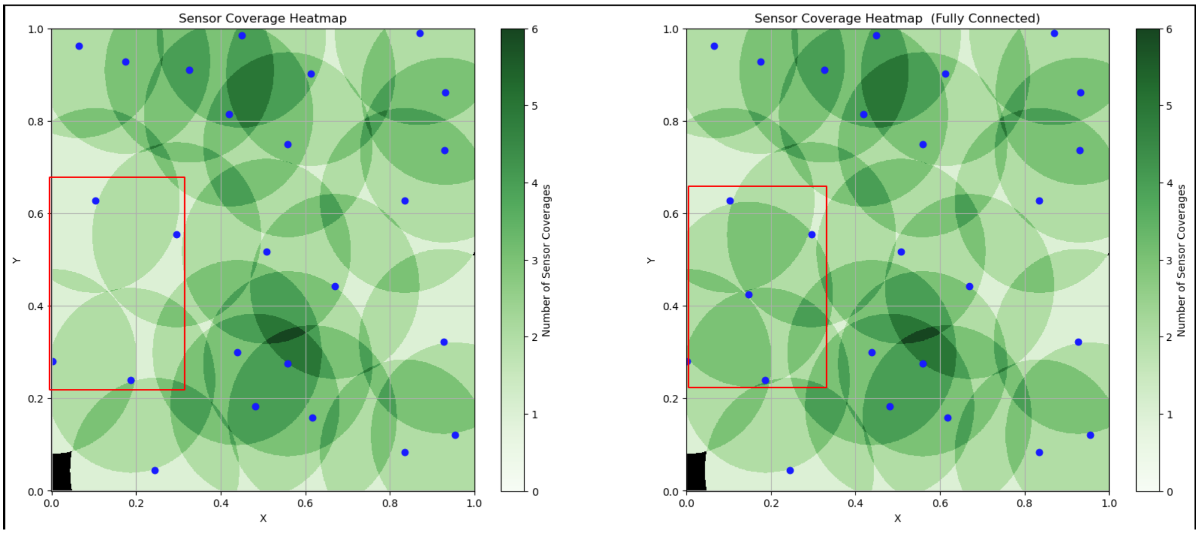
After Adding one node to the `nodes.list` now let's take a look at the communication and the sensor coverage diagram, on the left hand side is the `sensor coverage heatmap`, the darker the higher coverage rate, on the right hand side is the `communicate range diagram`, shows a fully connected diagram.





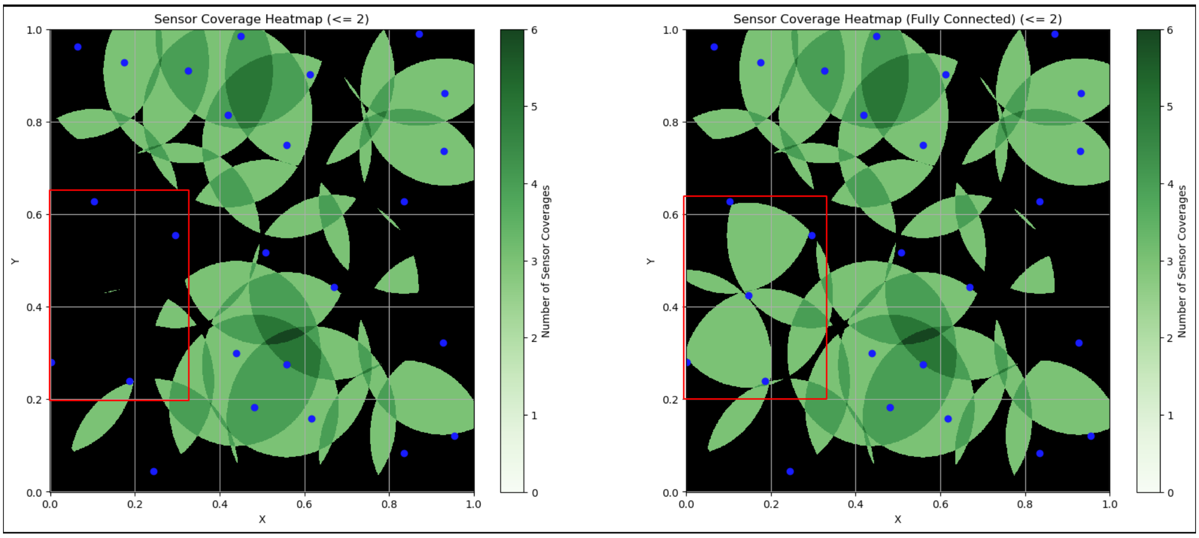
# Coverage of the Network

We have already demonstrate in the `Connectivity of the Network` that how we add one node `(0.147, 0.42525)` to make the network fully connected, and the reason of adding this node is that this node will largely enhance the coverage of this area, details can be shown in below diagram, the area with `0-coverage` is depicted with black colour!

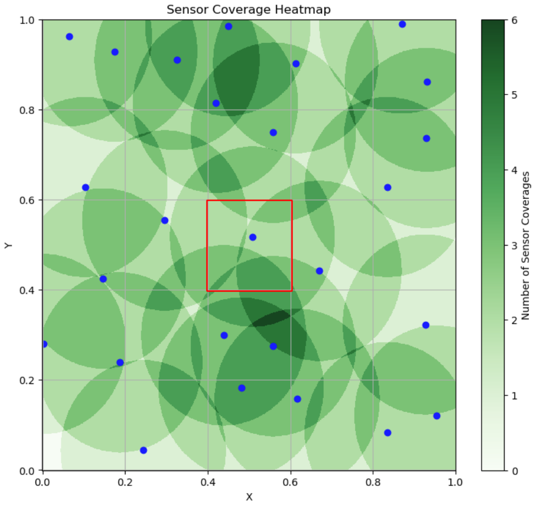


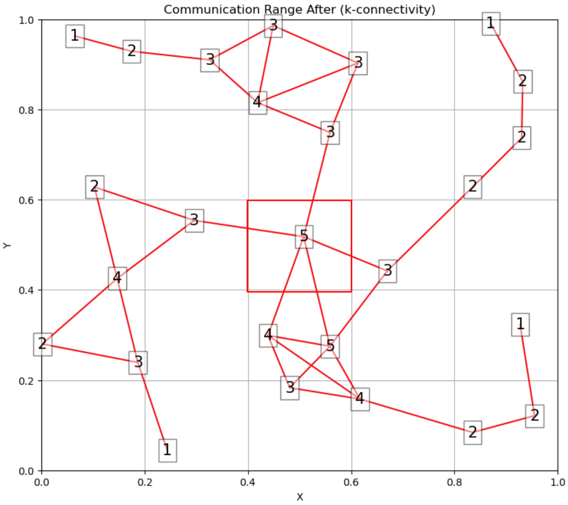
From above diagram we can tell that adding this node will not affect the area with `0` coverage, but will enhance the coverage of the areas depicted using red square. Further more, we can show the enhancement by adding below two diagrams, as we can see when make the area with the sensor coverage `<=2` coloured in black, we can see that by adding the extra one point we have successfully increased the robustness of the area designated by red squares.

From the diagram below we can see that ares coloured in green are better covered, and the area coloured in black are worse covered. The distribution of  coverage is worse close to the edge of the area.



Furthermore, if we take a look at the overall coverage diagram, we can see that after adding one extra node, the `k-coverage` of the network is still `0`, and when we blank more and more area by increasing the thresh-hold value of blacking, we can see that the coverage of this network has been located to specific areas that have higher density of sensor deployments. and we should notice that the connection part of the upper and lower part of the diagram will be separated once we lose this point from the network.



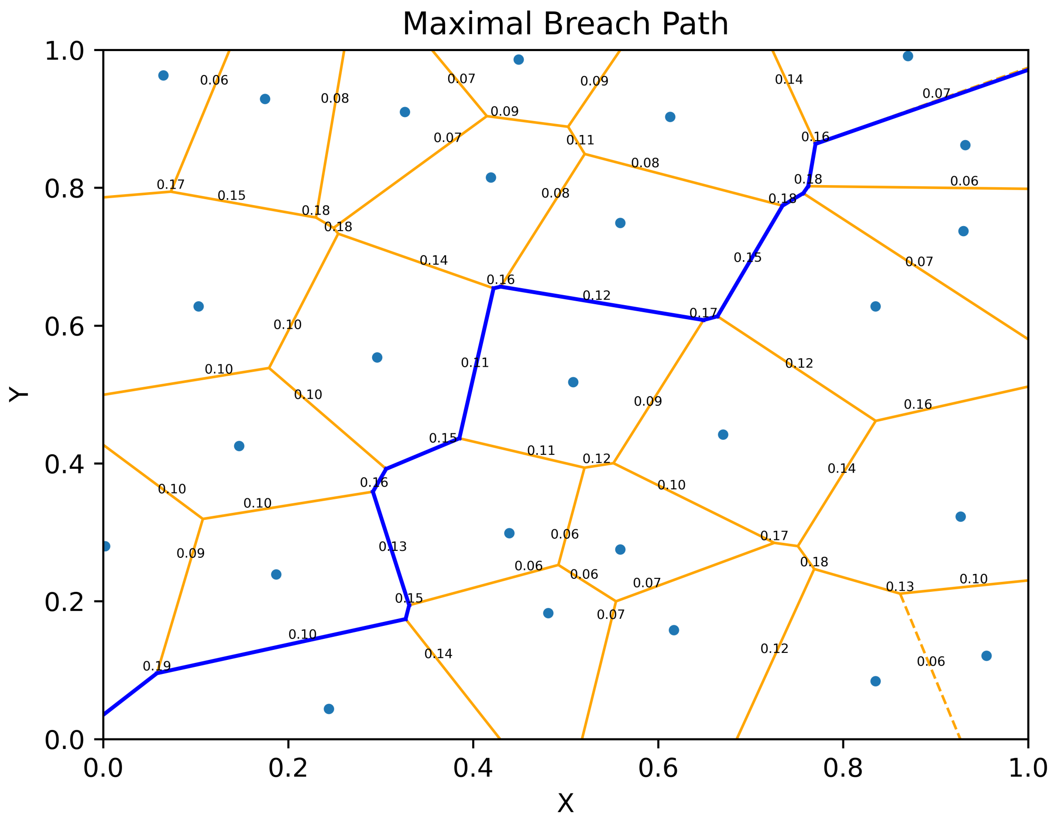


## Maximal Breach Distance & Maximal Breach Path

According to the requirements of the assignment, we have to calculate the maximal breach distance travelling from the coordinate `(0, 0)` to the coordinate `(L, L)`, and the cost of moving along the boundary of the network is `0`!

If we follow this rule, then the maximal breach distance would be `0` anyway! So I changed the question to be travelling from the point close to the `(0, 0)` to the point close to the `(L, L)`.

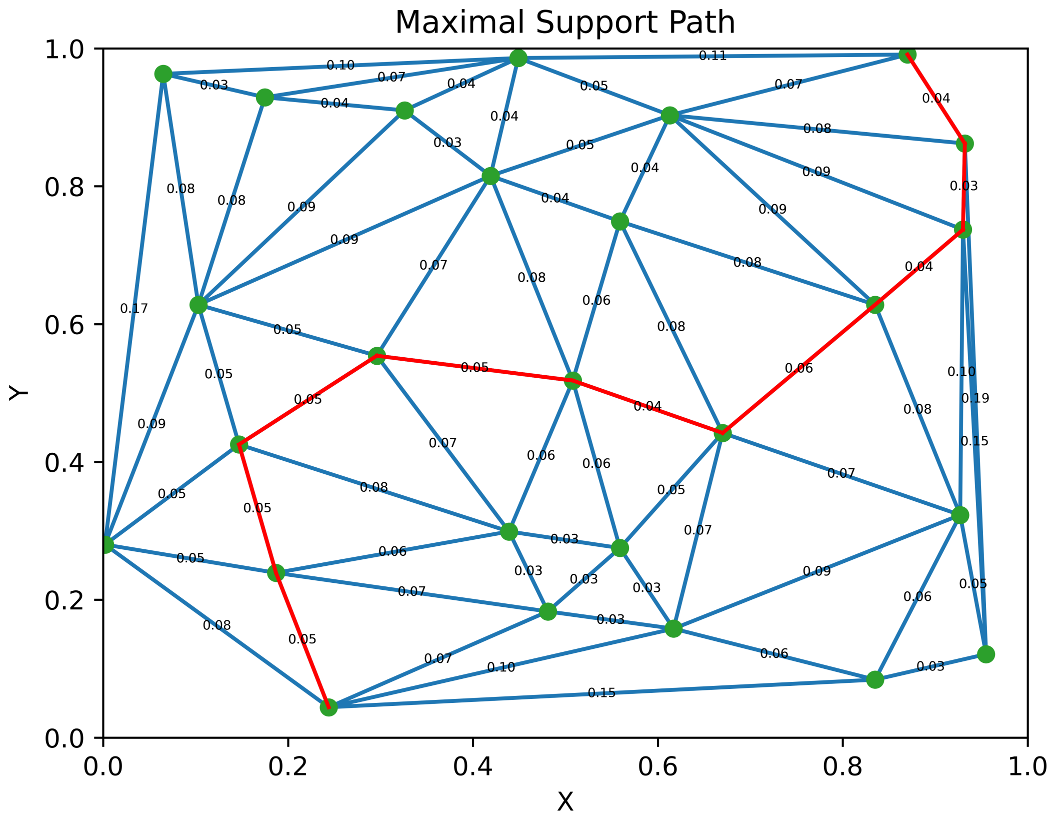
So the maximal breach distance is `0.07` , the very last segment of the path.



## Maximal Support Distance & Maximal Support Path

According to the requirement of the assignment, the starting point is the one closest to `(0, 0)` and the ending point is the one closest to `(L, L)`. Calculating the `cost` of every edges of the `Delaunay triangles` and display it on the graph.

The Maximal Support Distance is `0.06`, and the corresponding Maximal Support Path is shown in the graph below in red colour.



# Summary

Employ the technique like `k-connectivity` and `k-coverage` we can evaluation the network in terms of connectivity and coverage. Furthermore, when we abstract the network to do further analysis, `Maximal Breach Path` is a technique based on geometry and `Maximal Support Breach Path` is a technique based on topology, once you build up the model of the problem, you can always use `BFS` or `DFS` to solve specific problems!

All the diagrams used in this assignment is submitted in this GitHub repository: [https://github.com/dcheng69/Sensor_Network_and_Embedded_Systems/blob/dev-lab3/lab3/lab3.ipynb](https://github.com/dcheng69/Sensor_Network_and_Embedded_Systems/blob/dev-lab3/lab3/lab3.ipynb)