Jimi Ford
CSCI 351-01 Data Communications and Networks
Programming Project 3 Report

1. Program Inputs, Program Objective, Program Outputs

SimulationStation.java

This class is responsible for generating a network of randomly placed Space Stations in 3D space. It takes in a lower bound for the fewest amount of space stations to include, and an upper bound for the largest amount of space stations to contain in a network. Another parameter (or knob) that gets passed in to this program is the number of trials to run where a trial consists of randomly generating a network, and then extracting the relevant data from that network. Each trial runs Floyd-Warshall on the graph of space stations using the power required to send a signal from one station to another as the weight of an "edge" between two stations. The resulting 2D matrix contains the minimum total power needed for any station *A* to send a message to any other station *B* by asking stations along the shortest path to forward the message along until it reaches *B*. Every value above the reflexive 0 diagonal that is <u>not</u> infinite is averaged over all the trials for a given amount of space stations. This average appears on the plot for the average power needed for that single measurement. This process is repeated for every number of space stations the user requested at runtime. The program knows if a network is fully connected or not if it encounters an edge with infinite weight. A fully connected network has a way for every pair of stations to communicate and an infinitely weighted edge means there is no way for those two stations to communicate. Using this rule, the program keeps a count of how many connected networks it found and calculates the fraction of connected networks over the total number of trials for a given number of space stations. The objective of this program is to study random space station networks, how much power would be needed to communicate in them, and the probability of a space station network being connected given the number of space stations used within a finite 3D space.

PlotHandler.java

This program contains the necessary methods to write a plot file given a set of data and then display the results of that plot file. The input to this program is what plot(s) file to display and the output is a graphical representation of the given plot file(s). The objective of this program is to produce a visual or graphical representation of the results of running the above program.

2. Exact Command Line

SimulationStation.java

note: This program must be invoked by Prof. Alan Kaminsky's Parallel Java 2 library with the CLASSPATH environment variable set as per <u>Prof. Alan Kaminsky's instructions on this</u>.

```
usage: java pj2 SimulationStation <lower_bound_stations>
<upper_bound_stations> <increment_stations> <num_trials> <seed>
<file_prefix>
```

`<lower_bound_stations>` – Lower bound (inclusive) for number of vertices in space networks

`<upper_bound_stations>` – Upper bound (inclusive) for number of vertices in space networks

`<increment>` – amount to increment the number of space stations by for each simulation

`<num_trials>` – Number of space station networks to simulate

`<seed>` – the seed value for Prof. Alan Kaminsky's PRNG

`<file_prefix>` – Prefix for file output

PlotHandler.java

note: This program allows for any number (greater than 0) of plot files to be specified in the command line arguments.

```
usage:
java PlotHandler <plot-file-1> (<plot-file-2> <plot-file-3>... etc.)
```

`<plot-file-1>` – The plot file (generated by SimulationStation) to visualize in an X-Y plot

3. Source Code *(See Appendix B for project's source code)*

---

4. Based on data generated by running your simulation program(s), discuss this question: What is the probability that the network is connected as a function of the number of stations, and why is the probability behaving this way?

Looking at *Q5 Supporting Data*, we find an interesting trend in the probability of a network being fully connected depending on how many space stations are in it. The probability begins at about 16% and drastically decreases to 1%. This global minimum value of 1% occurs at 8 space stations. After 8 space stations, the probability of having a connected network shoots up quickly to 90% at about 60 space stations. From there we see the probability approaches 100% asymptotically and after about 175 stations, it is nearly guaranteed that we will have a

connected network. Observing Appendix A confirms that the frequency of 100% connectivity is especially prominent after 175 (or 174) space stations. Let us now look into why the probability of generating a connected graph is behaving this way.

To do this, consider the following oversimplification of our space station network model. This number line now represents the space we are confined to:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20

We will represent randomly placing a space station by randomly picking 1 of these numbers.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20

Say that these numbers (or space stations) can only talk to other numbers that are either equal to themselves or directly adjacent to them. This, of course, models the constraint of the power limitations each space station has. The above green square represents that we picked 7 as our first number. This means that if we want the next number we pick to be able to talk to the first number, we have to pick either 6, 7, or 8 on our next try.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20

Assuming our method for picking numbers is uniformly random, the chances of us picking one of these numbers is $3/20$. This means with this model we have roughly 15% chance of ending up with a network of numbers of size 2 that is fully connected. The reason it isn't exactly 15% is because if we pick either 1 or 20 for our first number, in each case we only have $2/20$ chance of picking a second number that is either equal or directly adjacent to the first number. If we now decide to pick a third number in hopes of maintaining a connected network, here are the possible outcomes. Green denotes what we picked and yellow denotes the possible numbers for pick #3 that would maintain network connectivity.

A)

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20

B)

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20

C)

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20

Each outcome, A, B, and C, has equal probability of happening. Recall that the second pick (represented by row 2) had roughly $3/20$ probability of being connected. To calculate the probability of row 3 happening (our third pick), we will look at each case individually.

Case A:

We picked 7 two times in a row. This is allowed since it is consistent with the original problem's constraint. Our original model allowed for space stations to occupy the same 3D space, so we will allow our space stations to occupy the same 1D space in this simplified model. We use $3/20$ for row 2's probability and multiply it by row 3's probability which is also $3/20$ to get $9/400$. This means that row 3 has only a 2.25% chance of happening. (This is not exact because we are ignoring the special case of picking 1 or 20 first.) We went from roughly 15% chance of being connected with 2 numbers down to 2.25% chance of being connected with 3 numbers.

Case B:

We picked 6 after 7. This now means for our third pick, we have to pick either 5, 6, 7, or 8 in order to maintain connectivity. This has a $4/20$ chance of happening. To obtain the total probability of maintaining connectivity after this third pick we must multiply row 2's chance by row 3's to get $3/20 * 4/20 = 12/400$ which is equal to 3% probability of being connected with 3 random picks. Again this is a rough estimate since we are deliberately ignoring the special case to make this explanation simpler.

Case C:

We picked 8 after picking 7. The probabilistic consequences are identical to Case B.

Since row 2 had equal probability of happening, we can average row 3's probability across each case: $(12/400 + 12/400 + 9/400)/3 = 11/400$ which is 2.75%.

From this, we see that going from 2 random numbers which have roughly a 15% chance of being connected, to 3 random numbers results in a drastic decrease in probability of connectedness. In fact, picking 4 numbers will have an even lower probability of yielding a connected network. The probability will continue to decrease to a global minimum where it will then begin to increase again. This is because the more numbers we pick at random, the more we saturate the playing field with numbers and the more numbers we have to choose from that will maintain a connected network. We can show this visually like so:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20

This is a highly probable outcome after picking anywhere from roughly 17~25 numbers. Note that the only 2 numbers left to pick that would yield a disconnected network are 19 and 20. From this, we can easily see that if we pick 1,000 numbers between 1 and 20 at random, we are almost guaranteed that the resulting network would be connected. The reason we are not guaranteed that the network would be connected is because since we are picking randomly, there is a **very slim** chance that we will never pick 19.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20

In this case, since 19 wasn't picked, we would end up with the number 20 all alone (as long as it was picked at least once) and disconnected from the rest of the numbers.

We can extrapolate this 1D simplification out to 3D in order to explain why the probability of being connected dips down at first as the number of stations increases. It is rather intuitive why the chances of being connected increase after a certain global minimum. The cube of 3D-space that we are limited to begins to become saturated with stations limiting the possibility of placing a station in a position where it would be too far from everyone else to maintain connectivity. This explains why the probability of being connected is fairly high after a certain point in our original 3D space network model and still contains some rare cases where the network is not connected even with many stations. It is easier to spot these rarities in the table included in Appendix A. The plot in Q5 appears to have 100% success in connectedness after about 125 stations.

---

5. Supporting Data

---

note: In order to save you time while grading, lower numbers of simulations are used. As a result, the plots produced are not extremely smooth but are <u>smooth enough</u> to convey the

Commands used:

This first command will run the simulations and save *.dwg* files as well as a *.csv* file with the prefix *"space"*. (See Appendix A for this table data)

```
java pj2 SimulationStation 2 200 1 1000 12345019 space
```

```
(expected runtime 4 min 39 seconds with hyperthreading on 2 cores)
(expected runtime 2 min 19 seconds with hyperthreading on 4 cores)
```

After generating the *.dwg* files, you must run the following command to visualize the plots. The probability displayed goes from 0 to 1; 0 being 0 percent chance and 1 being 100% chance.

```
java PlotHandler space-probability-connected.dwg
```
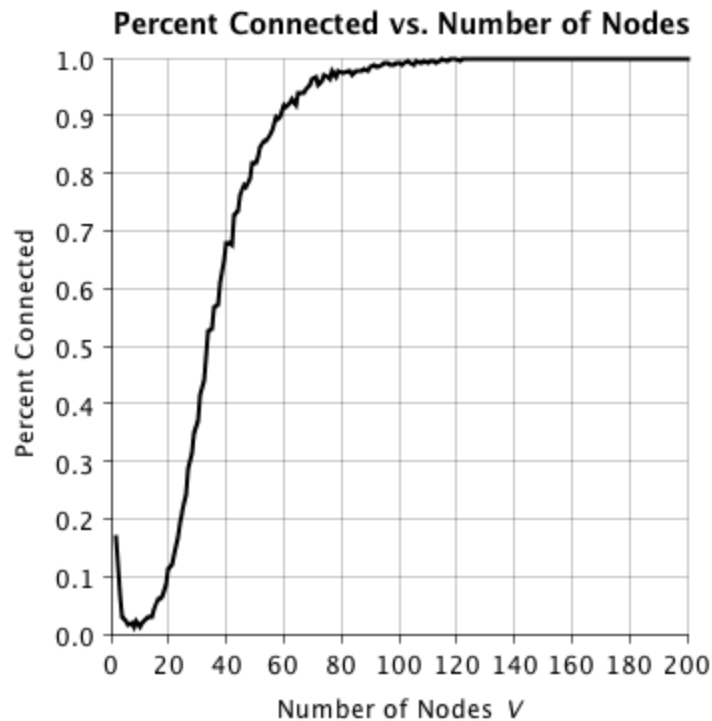

6. Based on data generated by running your simulation program(s), discuss this question: What is the average total power required to send a message from one station

to another as a function of the number of stations, and why is the average total power behaving this way?

Observing *Q7. Supporting Data*, we see a strikingly similarly shaped graph compared to the graphs from project 1. Project 1 was all about the average distance between nodes in a random graph. The power needed to transmit from one station to another contains the <u>distance</u> to the other station in the equation d^2. In project 1, the distance was a measure of # of hops. We can equivalently say that the distance was not a measure of # of hops, but rather equally weighted edges where every edge's weight was equal to 1. Scaling this weight by a factor of 40 million and adding kilometers on as a unit gives us a model comparable to our space station model. In project 1, we took the average distance from every pair of vertices where each distance factored in was the minimum distance. In this project, we used Floyd-Warshall to also determine the weight of the shortest path between every pair of vertices. We see that the average power needed starts at a global minimum, increases to a global maximum, and decreases asymptotically. This is because the average distance between stations is doing the same thing. If the average distance for a network increases, since the power needed to transmit a message is dependent on the distance, the average power will increase with this average distance. If the average distance for a network decreases, so will the average power needed to transmit these messages.

7. Supporting Data

---

note: In order to save you time while grading, lower numbers of simulations are used. As a result, the plots produced are not extremely smooth but are <u>smooth enough</u> to convey the general trend of the average power needed by each space station. The table results are found in Appendix A the "average_power" column.

Commands used:

We will use the results <u>generated from the previous run</u> and just view the plot file generated by it:

```
java PlotHandler space-average-power.dwg
```

Average Power vs. Number of Nodes
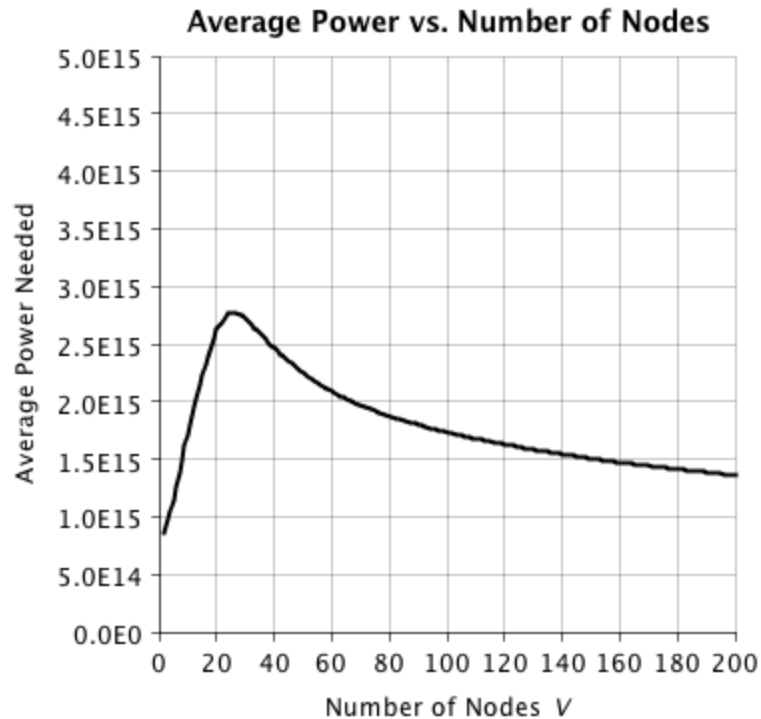
8. What I learned

This project was my favorite project we've done so far. This kind of problem is one I often fantasize about and it was fascinating to actually implement a program that modeled this problem and to observe the results for myself. I initially made predictions about what these results would look like: average power needed and probability of being connected. I was not too far off when I predicted that average power would decrease as more and more space stations were added, but I failed to correctly predict the global maximum that happens before the average power decreases. I correctly predicted that the probability of a network being connected would increase asymptotically to 1 (or 100%) but I failed to predict the initial dip in probability for very low numbers of space stations. I now have a heightened appreciation for the effects that the number of nodes have on networks and especially the effect average distance has on communication between nodes in a network. If I worked at NASA, I would now have a tool to accurately predict what would happen if we randomly placed stations out in space. As far as I understand, though, in a real network of space stations, they would need to be orbiting something. We would also not want to randomly place stations in orbit around celestial objects because as we saw, we have absolutely no guarantee that the network would be connected. It would cost us millions, if not billions of dollars to upgrade the capability of the space stations to reach further distances if we didn't plan accordingly and ended up with a disconnected network.

I would also be extremely mindful of calculating the average distance of the network because that is the determining factor of how much power it will take (on average) to communicate within the network.

Appendix A)

Table generated

num_stations	average_power	percent_connected
2	8.68E+14	0.162
3	9.46E+14	0.064
4	1.05E+15	0.024
5	1.13E+15	0.02
6	1.24E+15	0.011
7	1.32E+15	0.011
8	1.44E+15	0.01
9	1.55E+15	0.014
10	1.70E+15	0.013
11	1.80E+15	0.019
12	1.93E+15	0.025
13	2.02E+15	0.026
14	2.10E+15	0.03
15	2.24E+15	0.05
16	2.31E+15	0.054
17	2.45E+15	0.066
18	2.50E+15	0.078
19	2.55E+15	0.084
20	2.66E+15	0.11

21	2.65E+15	0.122
22	2.68E+15	0.126
23	2.74E+15	0.17
24	2.77E+15	0.19
25	2.74E+15	0.222
26	2.75E+15	0.233
27	2.75E+15	0.279
28	2.73E+15	0.306
29	2.71E+15	0.334
30	2.69E+15	0.38
31	2.70E+15	0.403
32	2.66E+15	0.451
33	2.64E+15	0.483
34	2.62E+15	0.517
35	2.58E+15	0.541
36	2.55E+15	0.583
37	2.52E+15	0.577
38	2.50E+15	0.622
39	2.46E+15	0.648
40	2.46E+15	0.69
41	2.43E+15	0.691
42	2.40E+15	0.702
43	2.39E+15	0.74
44	2.36E+15	0.746
45	2.33E+15	0.753
46	2.31E+15	0.78

47	2.30E+15	0.782
48	2.28E+15	0.795
49	2.26E+15	0.814
50	2.24E+15	0.822
51	2.22E+15	0.824
52	2.20E+15	0.861
53	2.19E+15	0.868
54	2.17E+15	0.861
55	2.16E+15	0.869
56	2.14E+15	0.871
57	2.13E+15	0.89
58	2.11E+15	0.891
59	2.10E+15	0.9
60	2.08E+15	0.915
61	2.07E+15	0.912
62	2.06E+15	0.923
63	2.04E+15	0.931
64	2.03E+15	0.922
65	2.02E+15	0.933
66	2.01E+15	0.945
67	2.00E+15	0.939
68	1.99E+15	0.943
69	1.98E+15	0.94
70	1.97E+15	0.962
71	1.96E+15	0.958
72	1.95E+15	0.954

73	1.94E+15	0.959
74	1.93E+15	0.965
75	1.92E+15	0.964
76	1.91E+15	0.957
77	1.90E+15	0.967
78	1.89E+15	0.965
79	1.88E+15	0.975
80	1.87E+15	0.977
81	1.87E+15	0.967
82	1.86E+15	0.976
83	1.85E+15	0.975
84	1.84E+15	0.971
85	1.83E+15	0.98
86	1.83E+15	0.975
87	1.82E+15	0.972
88	1.81E+15	0.982
89	1.81E+15	0.977
90	1.80E+15	0.986
91	1.79E+15	0.984
92	1.79E+15	0.989
93	1.78E+15	0.984
94	1.77E+15	0.992
95	1.77E+15	0.993
96	1.76E+15	0.991
97	1.75E+15	0.989
98	1.75E+15	0.992

99	1.74E+15	0.991
100	1.73E+15	0.988
101	1.73E+15	0.989
102	1.72E+15	0.994
103	1.72E+15	0.996
104	1.71E+15	0.994
105	1.70E+15	0.991
106	1.70E+15	0.995
107	1.69E+15	0.994
108	1.69E+15	0.992
109	1.68E+15	0.995
110	1.68E+15	0.994
111	1.67E+15	0.995
112	1.66E+15	0.997
113	1.66E+15	0.994
114	1.66E+15	0.998
115	1.65E+15	0.999
116	1.65E+15	0.997
117	1.64E+15	0.998
118	1.63E+15	0.999
119	1.63E+15	1
120	1.63E+15	0.998
121	1.62E+15	0.997
122	1.62E+15	0.998
123	1.61E+15	1
124	1.61E+15	0.999

125	1.60E+15	1
126	1.60E+15	0.999
127	1.59E+15	0.998
128	1.59E+15	1
129	1.58E+15	0.999
130	1.58E+15	0.998
131	1.58E+15	0.998
132	1.57E+15	0.998
133	1.57E+15	0.999
134	1.57E+15	0.998
135	1.56E+15	0.997
136	1.56E+15	0.998
137	1.55E+15	0.999
138	1.55E+15	1
139	1.54E+15	0.999
140	1.54E+15	1
141	1.54E+15	0.998
142	1.53E+15	0.998
143	1.53E+15	0.999
144	1.53E+15	0.997
145	1.52E+15	0.999
146	1.52E+15	0.999
147	1.52E+15	0.998
148	1.51E+15	1
149	1.51E+15	1
150	1.50E+15	0.998

151	1.50E+15	0.999
152	1.50E+15	1
153	1.49E+15	1
154	1.49E+15	0.999
155	1.49E+15	1
156	1.48E+15	1
157	1.48E+15	0.999
158	1.48E+15	1
159	1.47E+15	1
160	1.47E+15	0.999
161	1.47E+15	1
162	1.46E+15	0.999
163	1.46E+15	1
164	1.46E+15	1
165	1.46E+15	0.998
166	1.45E+15	1
167	1.45E+15	0.999
168	1.45E+15	1
169	1.44E+15	1
170	1.44E+15	0.999
171	1.44E+15	1
172	1.44E+15	0.999
173	1.43E+15	0.999
174	1.43E+15	1
175	1.43E+15	1
176	1.42E+15	1

177	1.42E+15	1
178	1.42E+15	1
179	1.42E+15	1
180	1.41E+15	1
181	1.41E+15	1
182	1.41E+15	1
183	1.41E+15	1
184	1.40E+15	1
185	1.40E+15	1
186	1.40E+15	1
187	1.39E+15	1
188	1.39E+15	1
189	1.39E+15	1
190	1.39E+15	1
191	1.38E+15	1
192	1.38E+15	0.999
193	1.38E+15	1
194	1.38E+15	1
195	1.37E+15	1
196	1.37E+15	1
197	1.37E+15	1
198	1.37E+15	1
199	1.37E+15	0.999
200	1.36E+15	1

Appendix B)

Source code