Lab7 (100 points)

Important: We will demo in class or online. For in class demos, we will go around the room. Be prepared to demo at the start of class. The submitted code will be used only to verify that you did not copy from others, to compile and re-run your program, to make sure you were indeed demonstrating your own code, and to grade for documentation of your code.

In this program we will start with Lab6. Everything that worked in Lab 6 should continue to work.

We are going to introduce the 2 new concepts. First, we will introduce the concept of movement. There will be a new move command added. When you get this message, you will update your location. These messages are NOT forwarded. You receive these messages regardless of the distance from the sender. The format for this message will be similar to lab 6, with the addition of the key-value pair move:x (where x is an integer number indicating the square to move to). Version will be 7. Minimally, the move command message will have toPort, fromPort, TTL, version, move

The second concept is a concept of optimization. If you get a message that is in range and NOT for you, after checking if the TTL is positive, you will then check to the send-path. You will NOT forward the message to any node already in the send-path. This is to reduce the traffic on the network. To validate that this works properly, the targeted receiver will not only print out the full message the first time it receives it (same as lab 6), but it will also print out JUST THE send-path when it receives a duplicate message.

Submit well-documented and well indented code along with a README file explaining how to run the program, and a makefile. Submit it using GitHub

The grading rubric is as follows:

- Program correctness and robustness (what happens if I give garbage input): 80%
- Coding style (comments, indentations, README, Makefile): 20%