Data Structures

Project Report

Group Members:

- (20i-0436) Shayan Faisal
- (20i-0871) Naik Ur Rehman
- (20i-0647) Shayaan Hasnain

Section A

Introduction

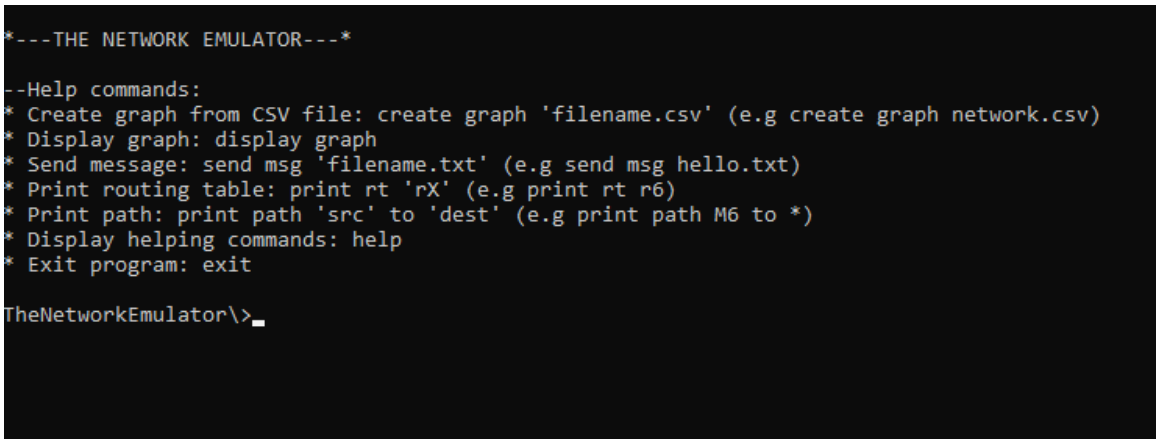
The goal of this project was to implement a network of routers and machines that are able to communicate through their links. Functionality required includes finding the shortest path using Dijkstra's algorithm, creating routing tables for routers, using Max heap priority queue for sending messages between machines, and other functionality like file handling.

Classes

The main classes we have implemented for this project are linked list, message structure, queues (fifo), Min heap for Dijkstra's algorithm, and Max heap priority queue for message forwarding, along with various other classes and structures for better organization.

Command Line Interface

For emulating the command-based approach of the program, we have designed the look to be similar to a windows terminal as shown below:



There is sufficient error checking present to allow the user to continue using the program instead of it exiting. These include conditions for incorrect commands, incorrect filenames, invalid lines in files, non-existing nodes in graph etc.

Read CSV file function

One of the major functions in the program is this one, as it allows user to specify a csv filename which can be traversed in order to find number of nodes for graph, names of all nodes, and their weights, all together forming an adjacency list representation of graph.

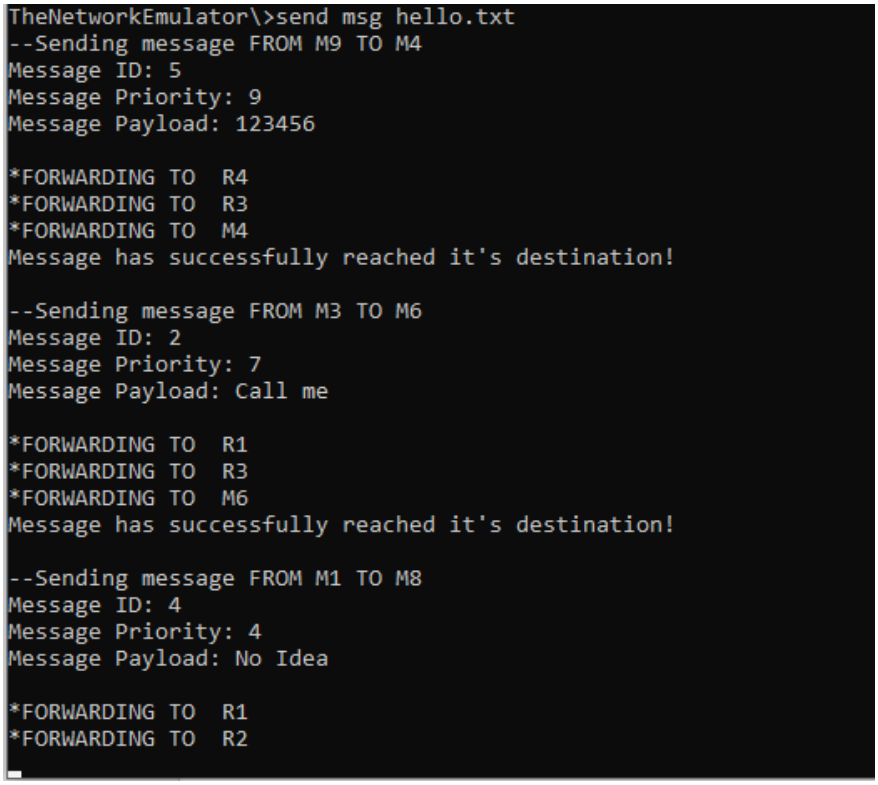
Dijkstra function

Another really important function in the program, it takes in the source node to apply Dijkstra on, along with a parent array (for path printing), and returns an array containing the shortest distances of this start node to each node in the graph. This can be modified to work for single node pair, single node to all nodes, all nodes to single node, or all nodes to all nodes, as demonstrated in the program. This function is also handy in creating routing tables for each router in the graph during graph construction from csv file. Complete working of this function is demonstrated and commented on in the code file.

Send Message function

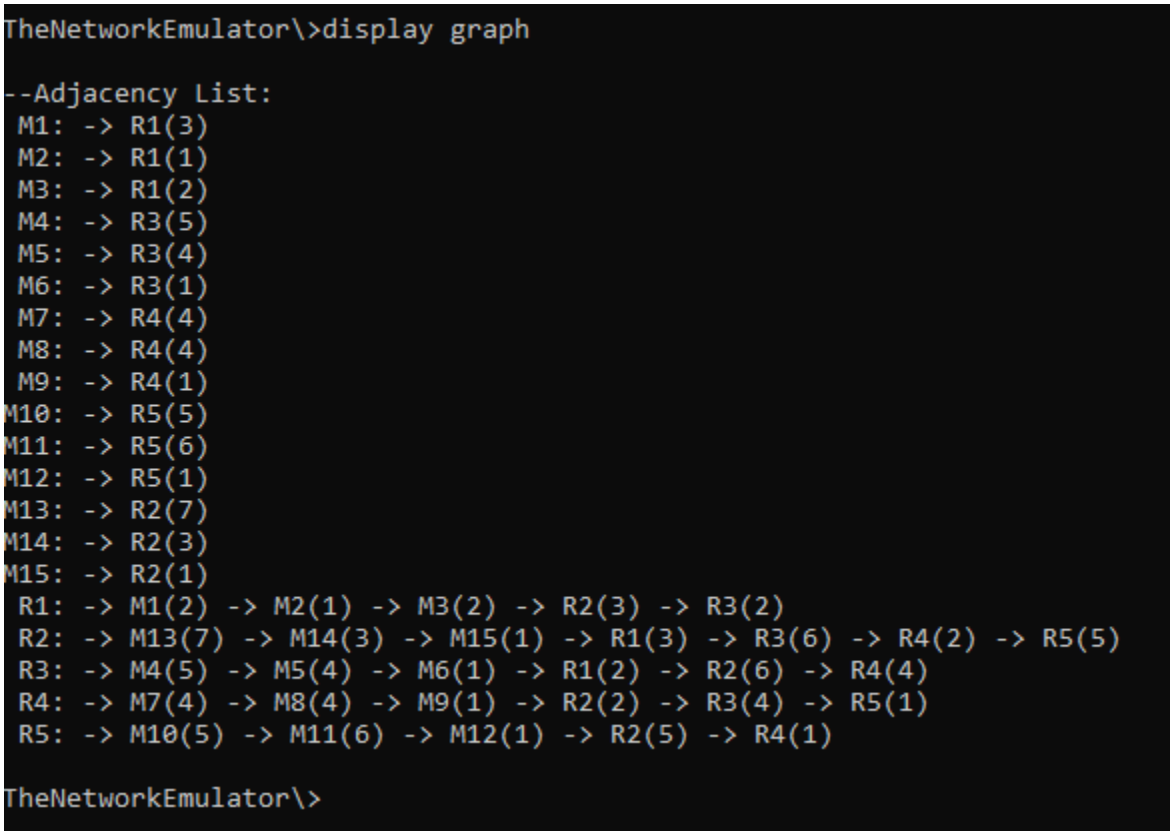
The working of this function is such that it takes a filename (e.g hello.txt) which contains all message parameters like id, priority, source, destination, and payload. Messages in the file are separated by newline, and thus read line by line to create objects of the message structure which are then inserted into the max heap priority queue.

Once all messages are read, message sending/forwarding process begins, which starts dequeuing the highest priority messages from max heap and using the routing tables created earlier, enqueues these messages into outgoing queue of current router. Then to simulate network lag, the “Sleep” function from windows.h library is used with a parameter of 1000 to provide a one second delay in between dequeuing message from outgoing queue of current router and enqueuing it to incoming queue of next router. As demonstrated in code, it displays the path taken by the message step by step, thus emulating how an actual network works like. This process continues for all messages until they’ve successfully reached their destinations.



Other various functions

There are many other functions present in the code, from helper functions to maintain consistency in command checking (like convertToUppercase and convertToLowercase), to display functions for various procedures like displayAdjacencyList, displayPath, displayRoutingTable etc.



Conclusion

Overall, we've implemented as much of the requirements of this project as possible, and after thorough testing, the results seem to be accurate. Only missing features are splay trees which could not be implemented and some minor commands for the program.