

# Distance-Vector-Planning-Routing

Computer Networking Protocols | CS 4470 - 01 | Professor Senhua Yu

```
=====
| ONLINE                                                                    |
|      SSSSSS EEEEEEE RRRRRR  V      V EEEEEEE RRRRRR                    |
|      S      E      R      R  V      V E      R      R                    |
|      SSSSSS EEEE  RRRRRR   V  V  EEEE  RRRRRR                    |
|              S E      R      R   V V  E      R      R                    |
|      SSSSSS EEEEEEE R      R   V  EEEEEEE R      R                    |
|                                                                    Port:# |
=====
```

## README

### DESCRIPTION

Distance-Vector-Planning-Routing implements a simplified version of the Distance Vector Routing Protocol . The protocol can run on 2 or more laptops behaving as servers using TCP. Each server runs on a machine on a predefined port number. The server accepts user available commands and outputs their forwarding tables along with link cost updates on user inputted time intervals and “step” command, at the same time.

### PREREQUISITES

Visual Studio Code

- Extensions
  - Debugger for Java by Microsoft
  - Extension Pack for Java by Microsoft
  - Maven for Java by Microsoft
  - Project Manager for Java by Microsoft
  - Test Runner for Java by Microsoft
  - Language Support for Java(™) by Red Hat

Java version “19.0.1”

Topology.txt data format

- <num servers> : total number of servers
- <num neighbors> : total number of edges/neighbors
- <server ID> <server IP> <server port> : Server id, corresponding IP, port #
- <server ID1> <server ID2> <cost>: Link cost between server id 1 and 2.

File Note: Constants.java

- Add server information, ip address and port, manually.

## CONTRIBUTIONS

---

Although we each contributed individually to this project.. Where we all contributed by sharing ideas while working on the same command or problem. For example, when it came to displaying the routing table for the display command, I (Karina) had trouble understanding how to retrieve information from the Hashmap. How we approached this problem was sharing 1 screen and sharing ideas.

Karina 🤔🔍🤔🍰😎(◡‿◡)

I worked on Server Responses/Output Format, and the following commands: display, disable, crash. Most of the commands that I worked on are similar in that they shared information and functionality, so when I completed one, the rest were easier to approach. For example, the crash and disable commands both close links so I just had to mirror the work and add additional requirements when needed.

Adrian

I created the foundation of the application. I created a thread to handle user input and a method with a switch case statement so my teammates can write code to handle user commands, `executeCommand` in `Server.java`. I also created a Thread in `Server.java`, called `incomingConn`, to handle any connections being made to the server and create a Node object for the incoming connection and store it in `List<Node> connectedToUS`. I also created `Node.java` to store server information, whether they are connecting to our server or being read from the topology file. I made `Node.java` a parent thread that has two threads that handles receiving and sending messages to `Server.java`. I also create threads to send and receive packets as well as packet commands.

Steven Wang

I worked on the topology file reading command and update command and created some functions to support these two commands. For the topology reading command, it finds the file with the matched file name and reads the text message. It splits and stores each information, number of servers and neighbors, server information, and link costs. For the update command, first, it checks if input servers exist and if it is the server that users are running on. Then, it checks if two servers are neighbors. After everything is checked out, it updates the routing table and stores the message.

Jonathan

I worked on creating the message format class which has the servers IP and port numbers with several setter / getter methods and a method that helps add to an arraylist an updated servers cost from their neighboring servers which will also increase the number of fields. I wasn't too familiar with java so I did have a harder time contributing but I still tried to help my group members solve bugs and find solutions for their code.

## SOURCES **y'-y**

---

- <https://www.geeksforgeeks.org/how-to-print-colored-text-in-java-console/>