**Computer Communications and Networks (ITCS 6166)**

**Project Part 2**

**Distance Vector Routing Protocol**

**Tejaswi Konduri (800965883)**

**Language used:** C#

**Introduction:**

In this project, we implement the distance vector routing protocol. The main idea is to find the minimum cost to go to each node in the system.

We maintain an input file where we keep the number of neighbors, the names of the neighbors, the port numbers of the neighbors. The input file basically acts as a forwarding table.

After every 15 seconds, we send out the current state of each node to all its neighbors. And, the receiving node will calculate the minimum cost to all the nodes, based on the received inputs from its neighbors.

We use Bellman-Ford equation to calculate the costs.

We can also edit the input file dynamically by updating the costs and the system will recalculate the costs and next hops based on the new values.

**Components of the project:**

1. Distance vector routing algorithm implementation:

* We use UDP to send-receive messages between neighbors.
* The program will act as both client and server. Generally, it will act as a server, and after a timeout of 15 seconds, it will switch to client mode, where it will send messages to its neighboring nodes.
* Each node must run on a separate command prompt instance.
* The program will take the port number and the input file path as inputs.
* Example: DVRouting.exe 9000 "C:\cn\GIT\a.dat"

1. A batch script file (execute.bat), which will automatically open a separate command prompt for each node and run the Distance vector routing algorithm implementation.
2. Input files:

* Each node will accept an input file (.dat file), which will contain information such as number of neighboring nodes, names and port numbers of the neighboring nodes.

1. Edge Value Changer utility:

* This utility is used for changing the cost of an edge between two neighboring nodes.
* Syntax: EdgeValueChanger.exe <source-node> <destination-node> <new cost>
* Example: EdgeValueChanger.exe a c 1.0

**Steps to Run the application:**

1. Build the project (DVRouting).
2. Copy the input file to the base folder.
3. Run “execute.bat” batch script.
4. We can change the cost of the edges dynamically by running the EdgeValueChanger.exe <node1> <node2> <cost>

**Input format:**

To DVRouting.exe:

Input Syntax: DVRouting.exe <UDP port #> <path of input file>

Example: DVRouting.exe 9000 "C:\cn\GIT\a.dat"

To EdgeValueChanger.exe:

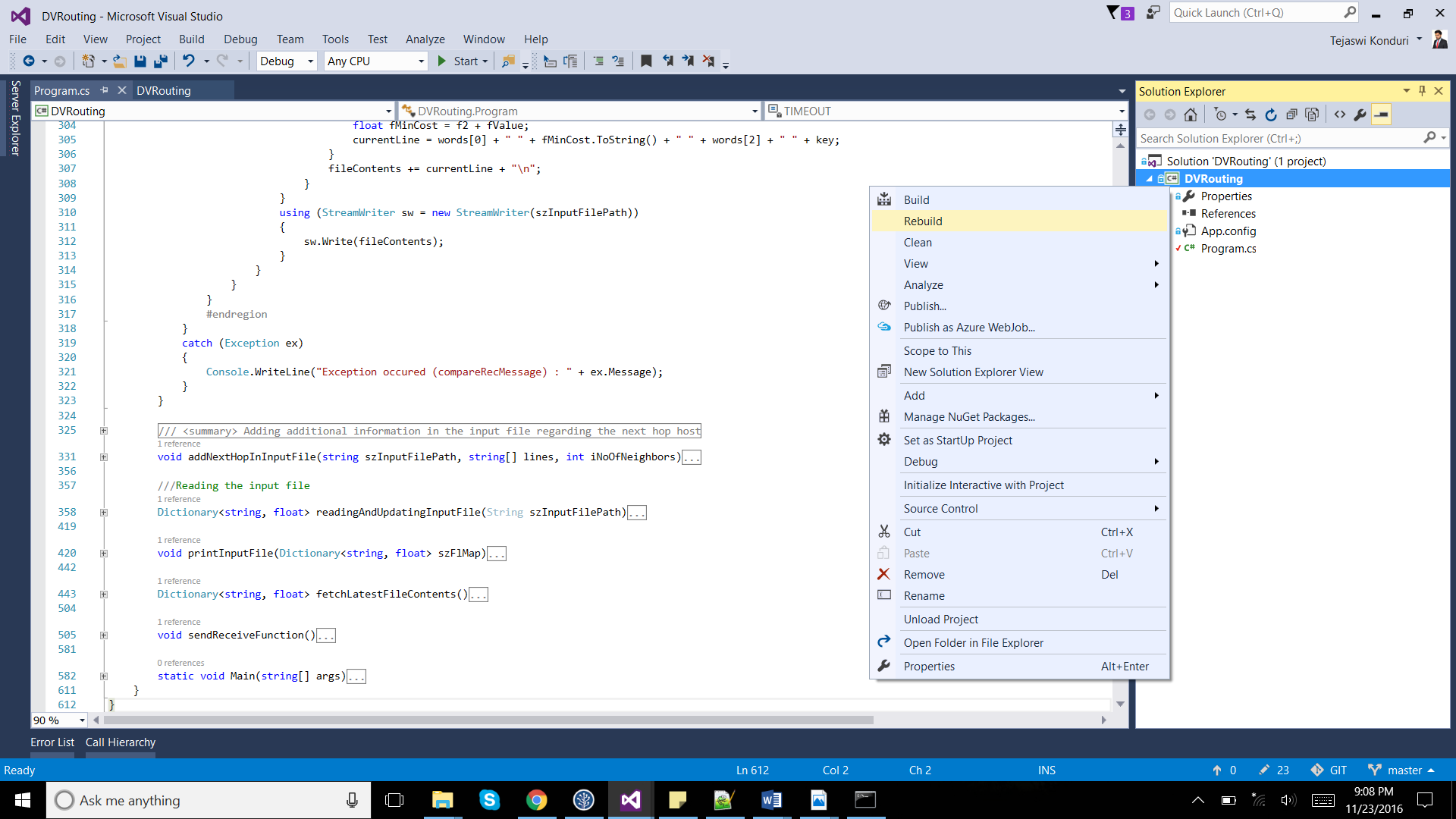
Input Syntax: EdgeValueChanger.exe <node1> <node2> <cost>

Example: EdgeValueChanger.exe a b 2.0

**Instructions to build and run the application in C#:**

**Steps** to run the application:

1. Unzip the project folder.
2. Copy the input files that need to be passed to each node as an input parameter to the base folder.
3. Build the DVRouting project (DVRouting\DVRouting .sln)
   1. Open the DVRouting.sln file in Visual Studio
   2. Rebuild the project from the Menu bar or Solution Explorer



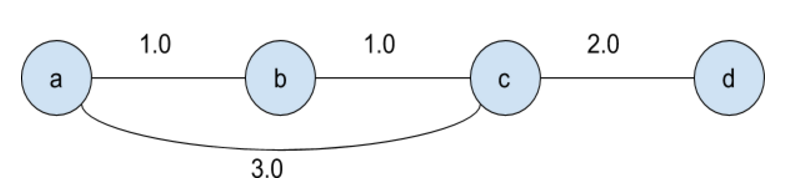
* 1. Open the path in the folder explorer to view the executable file (DVRouting.exe).

Example in our case: DVRouting\DVRouting\bin\Debug\

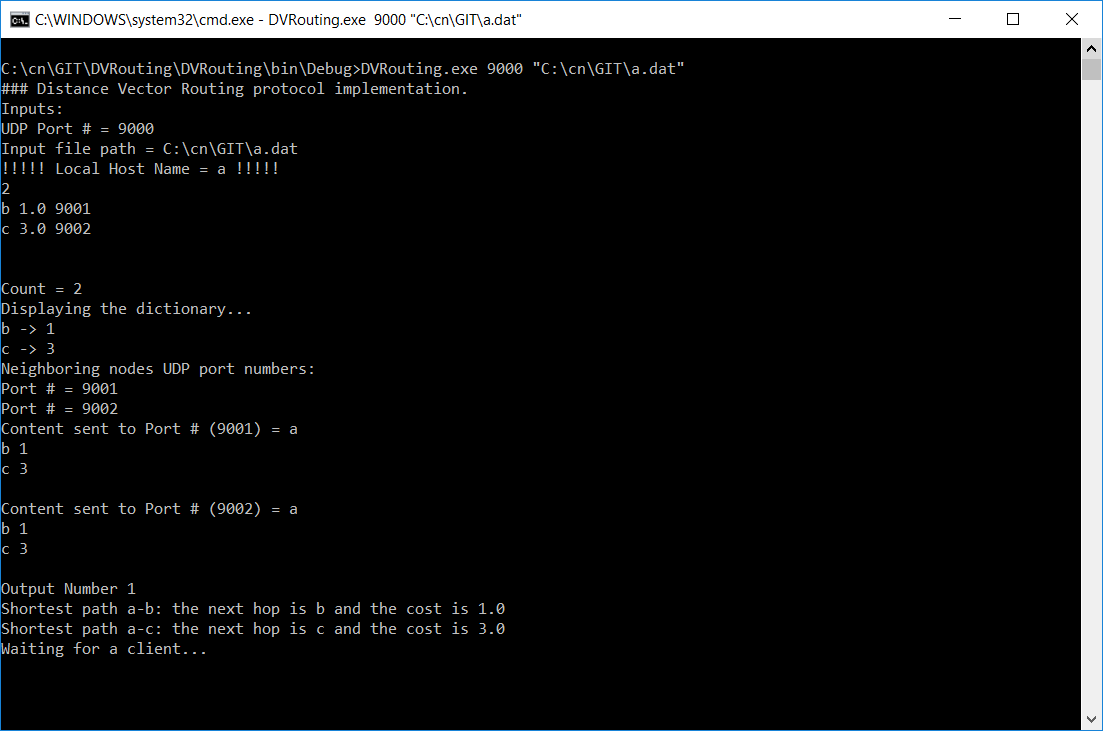
1. Run the executable (DVRouting.exe) in command prompt with appropriate command line arguments.
2. Or Run the execute.bat file, which will open multiple command prompt instances to run each node in each command prompt instance. But first you need to write the nodes details in the batch file, such as the port number and input file path.
3. Run EdgeValueChanger.exe to change the cost of the edges dynamically.

**Screenshots of the various operations:**

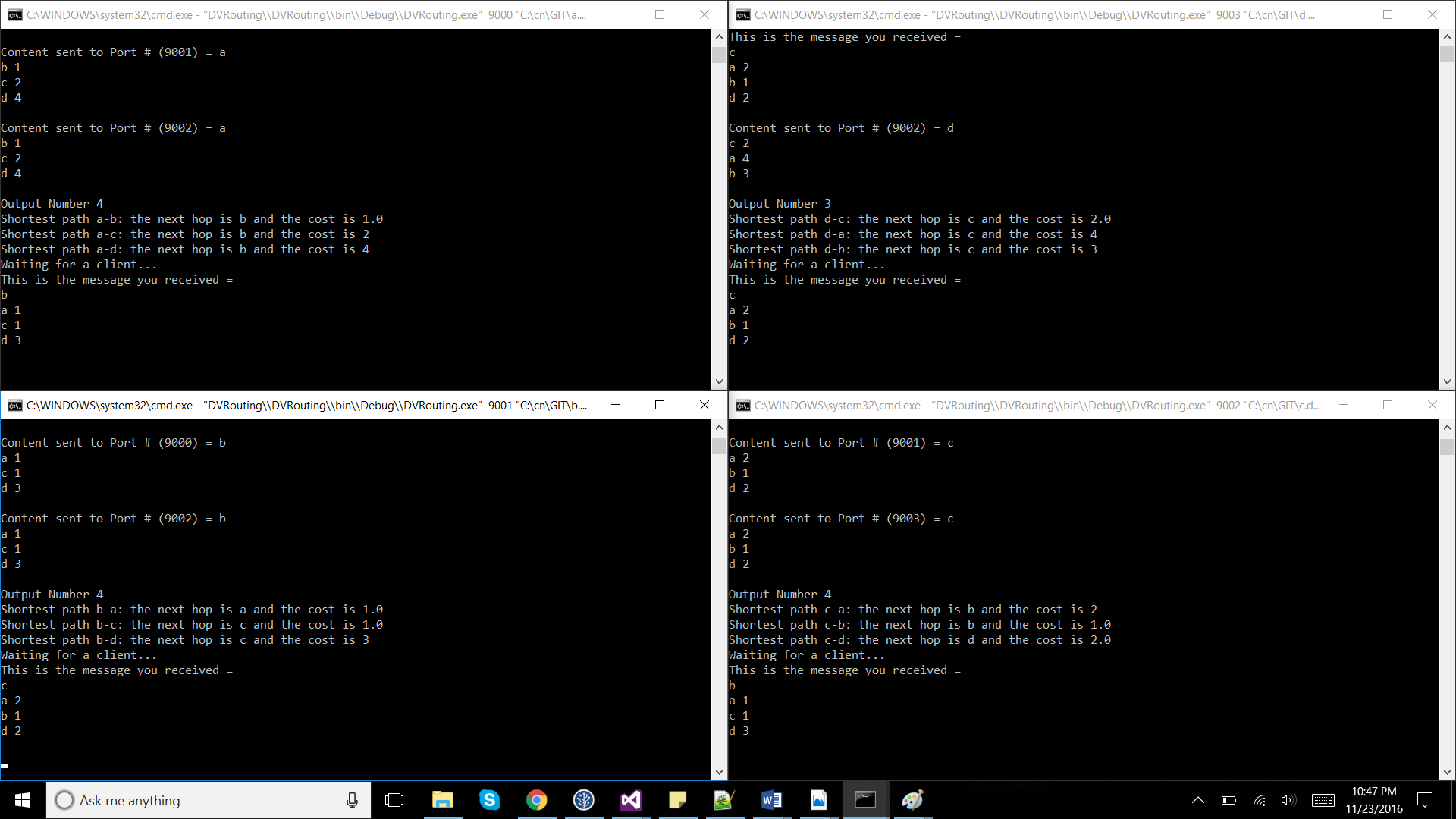
**Example:**

****

Running DVRouting.exe for node a:



Running DVRouting.exe for nodes a, b, c and d:



**NOTE:** There should be port numbers along with node and cost in the input file

Example: Table for node a (a.dat)

3

b 2.0 9001

c 5.0 9002

d 1.0 9003