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"
2) 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.
3) 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.
4) 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:

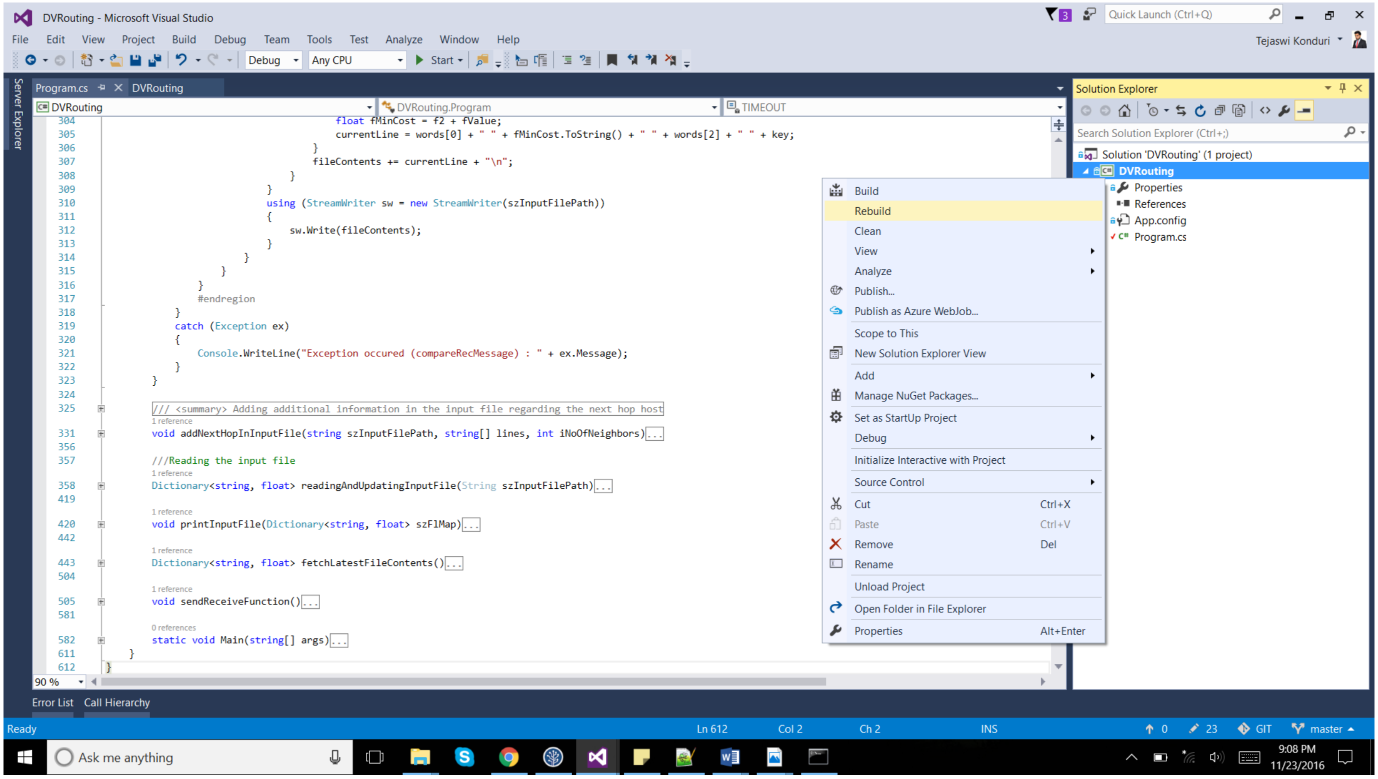
Input Syntax: EdgeValueChanger.exe <node1> <node2> <cost>

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)
    a. Open the DVRouting.sln file in Visual Studio
    b. Rebuild the project from the Menu bar or Solution Explorer



```
                                float fMinCost = f2 + fValue;
                                currentLine = words[0] + " " + fMinCost.ToString() + " " + words[2] + " " + key;
                            }
                            fileContents += currentLine + "\n";
                        }
                    }
                    using (StreamWriter sw = new StreamWriter(szInputFilePath))
                    {
                        sw.Write(fileContents);
                    }
                }
            }
        }
        #endregion
    }
    catch (Exception ex)
    {
        Console.WriteLine("Exception occured (compareRecMessage) : " + ex.Message);
    }
}

/// <summary> Adding additional information in the input file regarding the next hop host
1 reference
void addNextHopInInputFile(string szInputFilePath, string[] lines, int iNoOfNeighbors)...

///Reading the input file
1 reference
Dictionary<string, float> readingAndUpdatingInputFile(String szInputFilePath)...

1 reference
void printInputFile(Dictionary<string, float> szFlMap)...

1 reference
Dictionary<string, float> fetchLatestFileContents()...

1 reference
void sendReceiveFunction()...

0 references
static void Main(string[] args)...
    }
}
```

c. Open the path in the folder explorer to view the executable file (DVRouting.exe). Example in our case: DVRouting\DVRouting\bin\Debug\

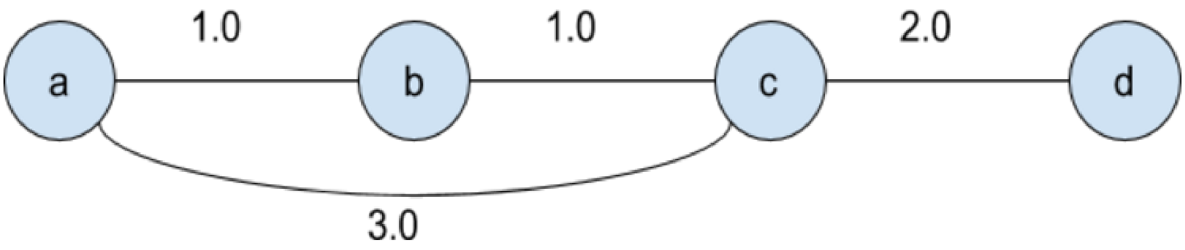
4) Run the executable (DVRouting.exe) in command prompt with appropriate command line arguments.

5) 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.

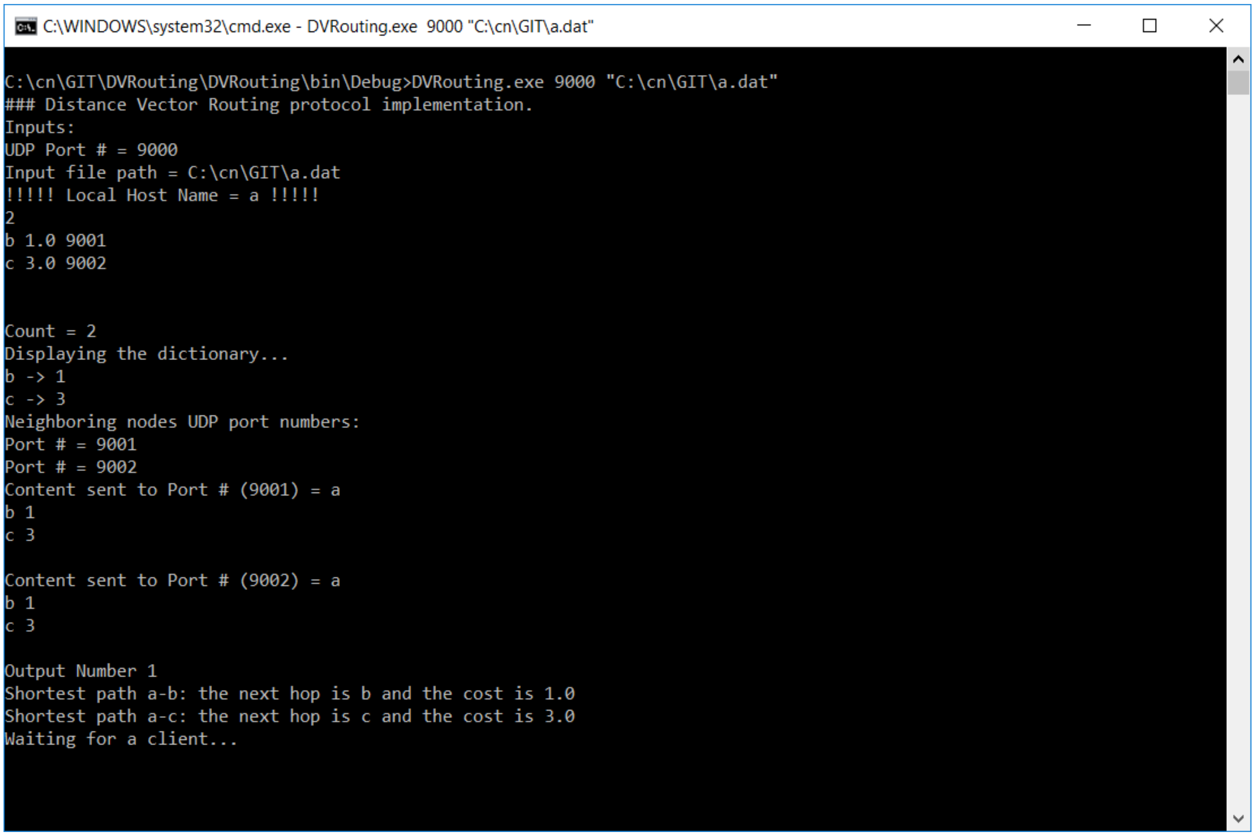
6) Run EdgeValueChanger.exe to change the cost of the edges dynamically.

Screenshots of the various operations:

Example:



Running DVRouting.exe for node a:


C:\WINDOWS\system32\cmd.exe - DVRouting.exe 9000 "C:\cn\GIT\a.dat"

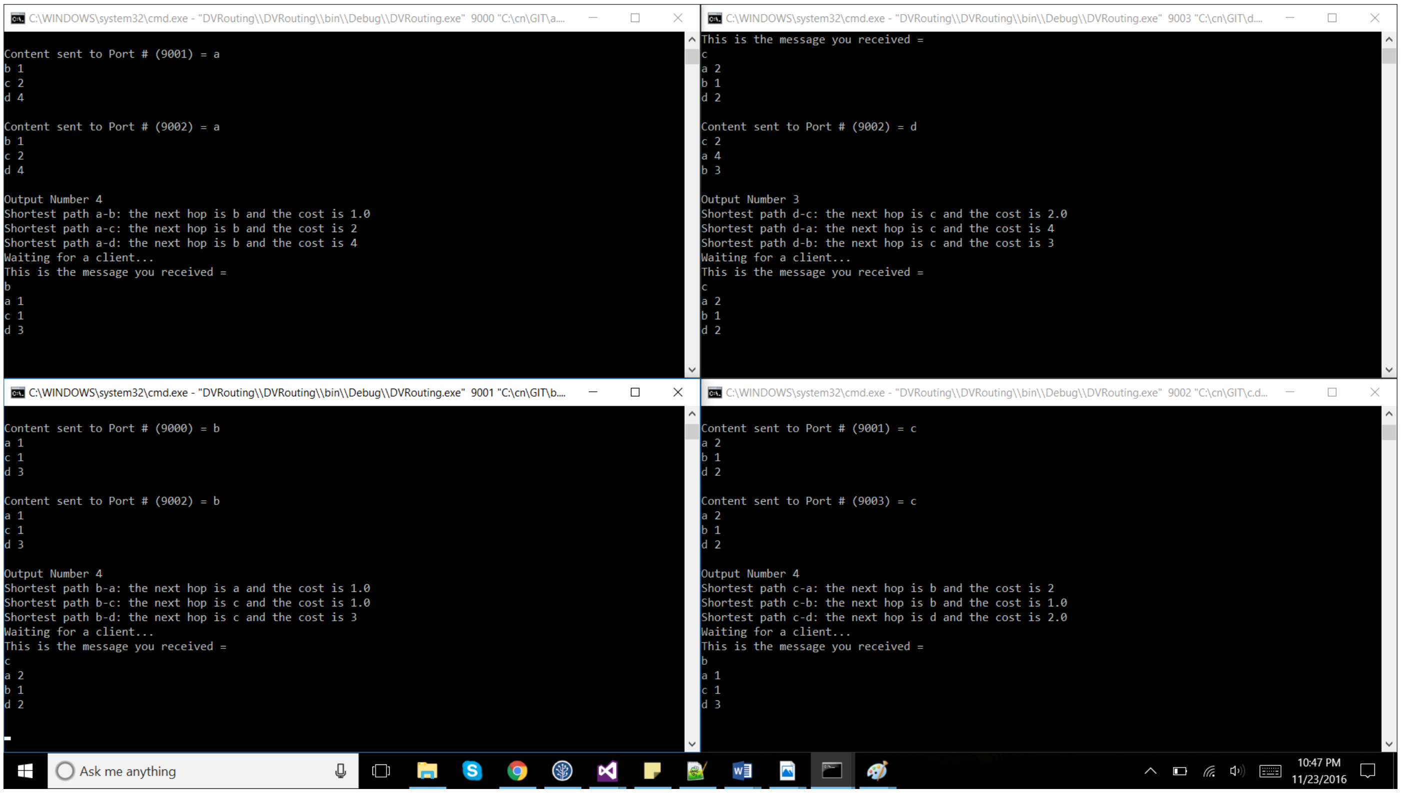
```
C:\cn\GIT\DVRouting\DVRouting\bin\Debug>DVRouting.exe 9000 "C:\cn\GIT\a.dat"
### Distance Vector Routing protocol implementation.
Inputs:
UDP Port # = 9000
Input file path = C:\cn\GIT\a.dat
!!!!! Local Host Name = a !!!!!
2
b 1.0 9001
c 3.0 9002


Count = 2
Displaying the dictionary...
b -> 1
c -> 3
Neighboring nodes UDP port numbers:
Port # = 9001
Port # = 9002
Content sent to Port # (9001) = a
b 1
c 3

Content sent to Port # (9002) = a
b 1
c 3

Output Number 1
Shortest path a-b: the next hop is b and the cost is 1.0
Shortest path a-c: the next hop is c and the cost is 3.0
Waiting for a client...
```

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


C:\WINDOWS\system32\cmd.exe - "DVRouting\\DVRouting\\bin\\Debug\\DVRouting.exe" 9000 "C:\cn\GIT\a....
C:\WINDOWS\system32\cmd.exe - "DVRouting\\DVRouting\\bin\\Debug\\DVRouting.exe" 9003 "C:\cn\GIT\d....
C:\WINDOWS\system32\cmd.exe - "DVRouting\\DVRouting\\bin\\Debug\\DVRouting.exe" 9001 "C:\cn\GIT\b....
C:\WINDOWS\system32\cmd.exe - "DVRouting\\DVRouting\\bin\\Debug\\DVRouting.exe" 9002 "C:\cn\GIT\c.d...
Ask me anything
10:47 PM
11/23/2016

```

Content sent to Port # (9001) = a
b 1
c 2
d 4

Content sent to Port # (9002) = a
b 1
c 2
d 4

Output Number 4
Shortest path a-b: the next hop is b and the cost is 1.0
Shortest path a-c: the next hop is b and the cost is 2
Shortest path a-d: the next hop is b and the cost is 4
Waiting for a client...
This is the message you received =
b
a 1
c 1
d 3
```

```
This is the message you received =
c
a 2
b 1
d 2

Content sent to Port # (9002) = d
c 2
a 4
b 3

Output Number 3
Shortest path d-c: the next hop is c and the cost is 2.0
Shortest path d-a: the next hop is c and the cost is 4
Shortest path d-b: the next hop is c and the cost is 3
Waiting for a client...
This is the message you received =
c
a 2
b 1
d 2
```

```

Content sent to Port # (9000) = b
a 1
c 1
d 3

Content sent to Port # (9002) = b
a 1
c 1
d 3

Output Number 4
Shortest path b-a: the next hop is a and the cost is 1.0
Shortest path b-c: the next hop is c and the cost is 1.0
Shortest path b-d: the next hop is c and the cost is 3
Waiting for a client...
This is the message you received =
c
a 2
b 1
d 2
```

```

Content sent to Port # (9001) = c
a 2
b 1
d 2

Content sent to Port # (9003) = c
a 2
b 1
d 2

Output Number 4
Shortest path c-a: the next hop is b and the cost is 2
Shortest path c-b: the next hop is b and the cost is 1.0
Shortest path c-d: the next hop is d and the cost is 2.0
Waiting for a client...
This is the message you received =
b
a 1
c 1
d 3
```

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