[PLACEMENT OF RSU IN HIGHWAYS]

REPORT OF PROJECT SUBMITTED FOR PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE DEGREE OF

BACHELOR OF TECHNOLOGY
In
INFORMATION TECHNOLOGY

By

SUBHANKAR DEBNATH

UNIVERSITY ROLL NO - 11700214071

GAURAV KIRTI

UNIVERSITY ROLL NO - 11700214034

VIKASH PATESHWARI

UNIVERSITY ROLL NO - 11700214084

UNDER THE SUPERVISION OF

ABANTIKA CHOUDHURY

[Assistant Professor]
[M.Tech. (Software Engg.)]
[RCC INSTITUTE OF INFORMATION TECHNOLOGY]

श्रमम् बिना न किमपि साध्यम्

AT
RCC INSTITUTE OF INFORMATION TECHNOLOGY
[Affiliated to West Bengal University of Technology]
CANAL SOUTH ROAD, BELIAGHATA, KOLKATA – 700 015
MAY – 2018

RCC INSTITUTE OF INFORMATION TECHNOLOGY

KOLKATA – 7OOO15, INDIA

CERTIFICATE

The report of the Project titled Placement of RSU in Highways submitted by Subhankar Debnath (Roll No.:11700214071 of B. Tech. (IT) 8th Semester of 2018), Gaurav Kirti (Roll No.:11700214034 of B. Tech. (IT) 8th Semester of 2018), Vikash Pateshwari (Roll No.:11700214084 of B. Tech. (IT) 8th Semester of 2018) has been prepared under our supervision for the partial fulfillment of the requirements for B Tech (IT) degree in Maulana Abul Kalam Azad University of Technology, West Bengal.

The report is hereby forwarded.

Countersigned by

Associate Prof (Dr). Abhijit Das	Abantika Choudhury
(HOD of Information Technology)	(Assistant Professor)
RCCIIT, Kolkata	RCCIIT, Kolkata

ACKNOWLEDGEMENT

I express my sincere gratitude to Abantika Choudhury of Department of Information Technology, RCCIIT and for extending their valuable times for me to take up this problem as a Project.

I am also indebted to other teachers for their unconditional help and inspiration.

Last but not the least I would like to express my gratitude to our entire department who helped me in their own way whenever needed.

Date: …………….. 2018

(Subhankar Debnath)
Reg. No.: 141170110166
Roll No.: 11700214071
B. Tech (IT) – 8th Semester,
Session-2014-2018, RCCIIT

(Gaurav Kirti)
Reg. No.: 141170110129
Roll No.: 11700214034
B. Tech (IT) – 8th Semester,
Session-2014-2018, RCCIIT

(Vikash Pateshwari)
Reg. No.: 141170110179
Roll No.: 11700214084
B. Tech (IT) – 8th Semester,
Session-2014-2018, RCCIIT

RCC INSTITUTE OF INFORMATION TECHNOLOGY

KOLKATA – 7OOO15, INDIA

CERTIFICATE of ACCEPTANCE

The report of the Project title Placement of RSU in Highways submitted by Subhankar Debnath (Roll No.:11700214071 of B. Tech. (IT) 8th Semester of 2018), Gaurav Kirti (Roll No.:11700214034 of B. Tech. (IT) 8th Semester of 2018), Vikash Pateshwari (Roll No.:11700214084 of B. Tech. (IT) 8th Semester of 2018) is hereby recommended to be accepted for the partial fulfillment of the requirements for

B Tech (IT) degree in Maulana Abul Kalam Azad University of Technology,West Bengal.

Name of the Examiner	**Signature with Date**
1. ……………………………………….	………………………………
2. ………………………………………..	……………………………..
3. ………………………………………	………………………………
4. ……………………………………….	………………………………

TABLE OF CONTENTS

ABSTRACT

Over the last few years, a lot of applications have been developed for Vehicular Ad Hoc NETworks (VANETs) to exchange information between vehicles. However, VANET is basically a Delay Tolerant Network (DTN) characterized by intermittent connectivity, long delays and message losses especially in low density regions [1]. Thus, VANET requires the use of an infrastructure such as Roadside Units (RSUs) that permits to enhance the network connectivity. Nevertheless, due to their deployment cost, RSUs need to be optimally deployed. Hence, the main objective of this work is to provide an optimized RSUs placement for delay-sensitive applications in vehicular networks that improves the end-to-end application delay and reduces the deployment cost. In this project, we first mathematically model the placement problem as an optimization problem. Then, we propose our own technique that places RSUs only in useful locations and allows both vehicle-to-vehicle and vehicle-to infrastructure communication: (i) the first step is comprehensive study that looks for the RSUs candidates locations based on connectivity information, and (ii) the second step uses genetic algorithm and Dijkstra algorithm to reduce the number of RSUs based on the deliverance time requirement and the deployment cost.

INTRODUCTION

In the latest years, the innovative progress in both transportation and communications technologies made information exchange possible on roads using smart cars [2]. Through communication, Vehicular Ad Hoc Network (VANET) is created. VANET is composed of a set of high mobile nodes, which results in a frequently changing network topology with intermittent connectivity. In VANET, there are two possible ways to communicate: vehicle to vehicle (V2V), vehicle to infrastructure (V2I) or both called V2X [3]. V2I communication capabilities have a huge impact on the network reliability and information exchange [2]. To enhance the network coverage and hence the quality of services, the infrastructure must be optimally installed. Infrastructures deployment is not trivial and needs intensive investigations. In this project, we focus on road-side units (RSUs) placement. RSUs act as wireless access points. We studied and analyzed the existing works to find out their limitations. Some works focus on highways scenarios and propose to put RSUs in intersections to communicate information to a large number of cars; some others suggest distributing RSUs randomly or uniformly on the road. In addition, most of related work limit the communication to the RSUs only instead of using both V2I and V2V communications. Consequently, initial RSUs locations are not optimized to support frequent network disconnection. Hence, they are not adapted to delay-sensitive applications as the delay constraint is not respected. In this work, we propose a location solution to place the RSU. The later is a two-steps solution that optimizes RSUs deployment for vehicular networks applications with delay sensitive constraints. In the first step, we develop an algorithm to find the most relevant RSUs candidate locations based on network connectivity and hence avoiding the useless positions with high density where vehicles can communicate with each other’s without using any additional infrastructure.

PROBLEM DISCUSSION

10.1 PROBLEM DEFINATION

We want to find the ideal place where should we place our RSU(road side unit) in the highways. to find that we must know the most no. of vehicles passed over a part of the highway i.e. vehicles per road part density

10.2 PROBLEM SOLUTION

Our approach was very simple yet very calculative. We divide highway into segments with keeping junctions. We enter the details of the vehicles passed over a junction and then comparing each junction with other we calculate where the density of vehicles per segment is more and place the RSU(Road Side Unit) respectively on that junction

10.3 OVERVIEW

- First we are asked the number of junctions.
- After entering the number of junction it will asked for number of segments for which we want to divide the area.
- After entering the number of Segments it will asked to enter the name of junctions
- After entering the name of junction it will asked to enter the number of vehicles we want passed from this junction
- Now this process will keep on repeating till the junction we have entered first
- Now it will calculate according to our algorithm.
- Hence the output will be in the form junctions where RSU to be placed first.

Advantages of Project

- It is very easy to implement and can easily calculate the number of vehicles per junction.
- The implementation cost is less.
- This is less time consuming.
- This can be use in simulators.
- As this is a research based project so no hardware implementation.
- Design of various network and scenarios is made possible.

For this modification of Routing protocol is become easy.

LITERATURE SURVEY

Road-side units positioning is a covering location optimization problem. In covering location problems, the goal is to find the optimal positions to cover all the clients (vehicles), while taking several constraints into account such as 2015 12th Annual IEEE Consumer Communications and Networking Conference (CCNC) 978-1-4799-6390-4/15/$31.00 ©2015 IEEE 127 deployment cost and applications quality of services requirements. A plethora of work is done to cover the field of RSUs optimized siting. In the research the authors aim to place RSUs optimally in order to enhance connectivity. They consider intersections as best locations in urban scenarios since the density in intersections is usually higher and information can propagate in all directions. In [6], the authors aim to improve data dissemination in an highways area. They model the location problem as a Maximum coverage with Time Threshold Problem (MCTTP [10]). To optimally deploy RSUs to cover the maximum vehicles in the road, they use a genetic heuristic to solve the MCTTP and get the best RSUs placement. In [7], the goal is to find the optimal positions for RSUs to cover the vehicles in urban roads and reduce the delay of safety messages propagation. The authors consider intersection positions as potential locations to deploy RSUs. They mathematically model the problem and use two methods to resolve it: (i) analytical Binary Integer Programming (BIP) [11] to find best time to broadcast in the whole area, and (ii) Balloon Expansion Heuristic (BEH) method to find best time over each route. In [8], the authors propose a cost-efficient RSUs deployment scheme. They aim to update security certificates in urban area within short delay using the deployed RSUs. In this scheme, they suppose that each vehicle can communicate with RSUs in bounded driving time whatever its actual position, and the extra overhead time used for adjusting routes to update certificates. They model the problem as a setcovering problem and use the polynomial-time approximation algorithm called “Greed Set Cover” [12] to solve it. In [9], the authors propose an RSUs siting strategy for file downloading in the case of urban scenario. The main goal is to guarantee file downloading success ratio and delay requirement with the lowest deployment cost. They also use intersections as RSUs initial placement and consider only V2I communications. To improve file downloading success ratio, they first model the inter-meeting time between vehicles and RSUs as a time continuous homogeneous Markov chain with two states (disconnected and connected to RSU). Then, they deduce the objective function that relates RSUs deployment and file download ratio success and delay. After that, they model the road network as a weighted undirected graph where each edge represents the average passing time on the corresponding road. Finally, they use the depth-first traversal algorithm to get optimal placement to deploy RSUs. However, the authors in [6][7][8][9] consider only V2I communications, which introduce a large delay when no car is available to drive toward the RSUs. In [13], the authors study the problem of deploying gateways to provide the required coverage while minimizing the deployment cost in urban area. They formulate the problem as a Maximum Coverage Problem (MCP) [14].They extract and analyze traffic data over several hours in a given area. They divide the studied area in uniform zones and count the number of vehicles entering and leaving each zone for each time unit to obtain the distribution of transition probability between two zones. The later is used to find the candidate zones to deploy gateways. Finally, they develop their own heuristic algorithm called “MobGDeploy” to find the optimal gateways locations. However, the authors try to cover the maximum clients by choosing to put gateways in the zones with high density; so the solution does not take into account the delay requirement especially in low-density cases. Such a solution is also not efficient to transmit safety messages for example. In [15], the authors aim to optimize the travel time between neighbors landmarks. They try to optimally find the best RSUs positions in urban area in order to efficiently aggregate traffic information on the road and use it to improve route planning. They use genetic algorithm to find best positions. For each iteration of the genetic algorithm, they use NS2 simulator to evaluate the current solutions and to estimate the required travel time by simulation. However, they choose a random candidates deployment positions, which increase the complexity of the solution and does not take into account density information. In [17], the authors' goal is to study the size of the gaps between RSUs to improve data collection and delivery ratio in a highway scenario. They initially distribute RSUs along the road equidistantly and formulate the distance upper bound between RSUs. Then, they use Omnet++ simulator [18] to evaluate the considered system under realistic settings to find the best gap between every two successive RSUs. In [19], the authors' goal is to minimize the average reporting time of information to a given RSU. To effectively collect data in highway scenario, they propose to use a uniform initial RSUs distribution (each twice radio range distance). To reduce the deployment cost and get the best RSUs placement, they develop a heuristic based on balloon expansion method. However, the authors in [17] and [19] propose their solution only for a single road. In addition, the uniform distribution is not the best initial distribution, as it does not take into account the density of the cars in the road and increase the delivery delay of information. In [20], the authors model the network connectivity using a fluid model and a stochastic model. They propose RSUs assignment as an application of their model. The fluid model is used to compute the network density. The stochastic model takes into account the vehicles’ random behavior. This model is different than the Poisson-arrival-location model (PALM) described in [21]. In fact, this model is designed for urban scenarios; it takes into account traffic light and interactions between vehicles. It allows determining the degree of connectivity in a given road. Then, it assigns the RSU to enhance the connectivity where the number of connected nodes is lower than a threshold. However, the authors find the best positions to place RSUs although they do not optimize the number of RSUs. In [22] the authors study the connectivity in VANETs based on results of percolation theory. They consider cars density, the rate of equipped vehicles, and communication coverage to study the distribution of isolated nodes and the impact of putting the RSU in intersections. The authors show that RSUs placement in crossroads does not impact significantly the proportion of isolated cars. From the related work, we can observe that the problem of RSUs deployment depends on the application. Many works consider only V2I communication i.e. the vehicle must be in the RSUs’ radio range to communicate messages. In addition, the authors choose to use intersections in the case of urban scenario and uniform distribution in the case of highway scenario, which are not the best RSUs positions. Consequently, in our work, we deal with RSUs deployment in both urban and highway scenarios for delay-sensitive applications. We use V2X communications to enhance the network connectivity and reduce unnecessary infrastructures.

SRS(Software Requirement Specification)

RSU Placement is the starting job of creating VANET architecture
next we have to build the algorithm for RSU Placement
Next we have to build the C code of the algorithm.
From this C code we have to get the junction name where maximum no. of cars are passing.
next we have to find the shortest path by the Dijsktra Algorithm in where we place the RSU.

Procedure for setting up algorithm and code :

- First we have to find the problem
- Next create the Algorithm
- We have to create the C code for the algorithms.

Software Needed

- CodeBlocks 17.12
- Wndows 10

Hardware Needed :

- 512 MB RAM
- Intel Pentium Dual core Processor

PLANNING

The planning process included the following steps:

First Step:

a) We discussed on the basic requirements of software and other materials for a start

b) A blue print was made keeping in mind the requirements.

c) A certain analysis was done based on the blue print generated.

d) The overall cost was set up .

e) We started to find algorithms of this project from the previous Research papers to modify and get some Idea about our project.

f) Further requirements were checked.

g) Building algorithm is in under progress.

Second Step:

a) Next we have to check the algorithm finally and then find the modification area.

b) Start coding.

c) after that we have to check the result.

d) send the total project to the Testing panel.

DESIGN/ALGORITHM

RSU PLACEMENT USING SEGMENTATION

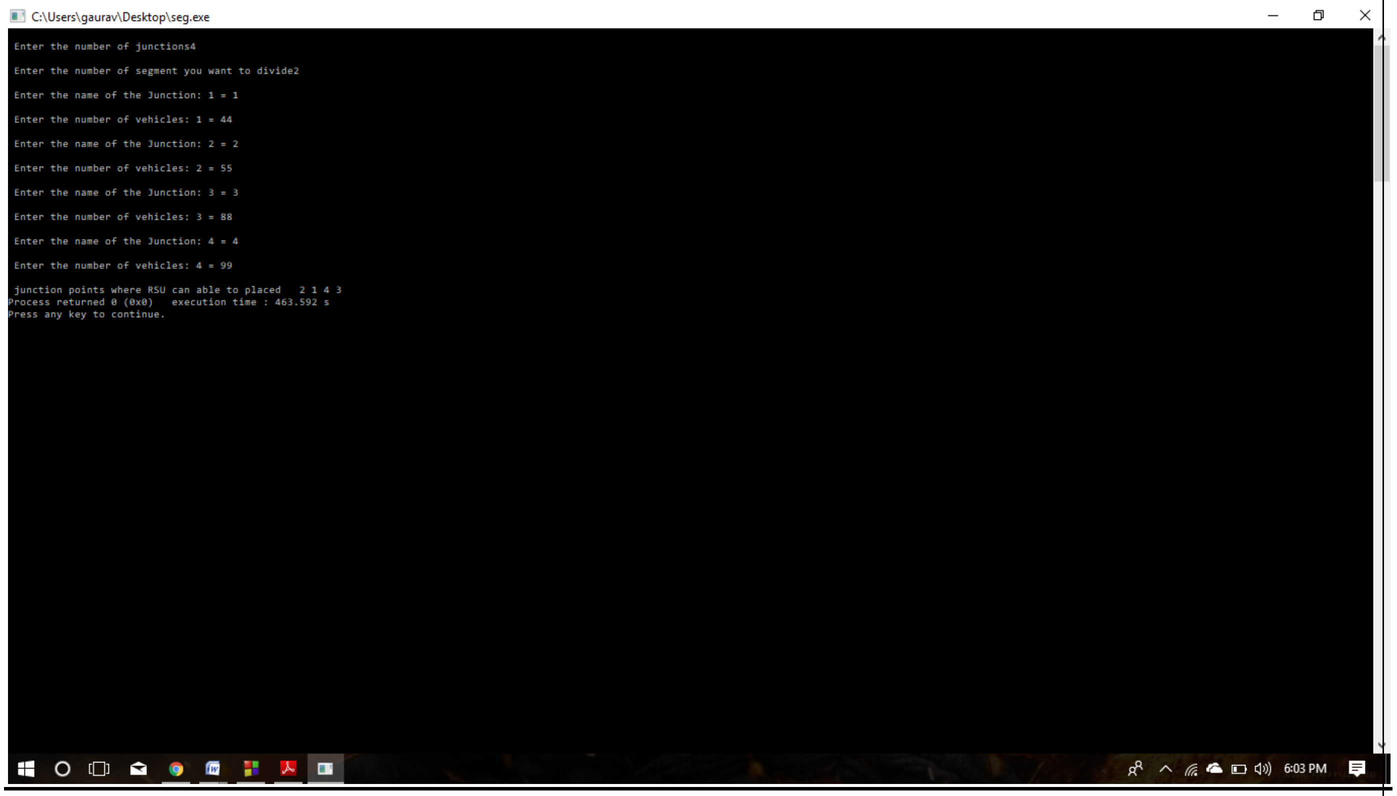
```
Steep 1-  Initialize
              • Array of junctions W
              • No. of RSU to be placed.
              • No of segments want to create
Step 2:  for i to a:
          Enter the elements of junctions of array
Step 3:  for i to a-1:
          For j= i+1 to a:
              If(W[i] < W[j])
                  Swap;

Step 4: for i= a to n-1:
          For j=i+1 to n:
              If(W[i] < W[j])
                  Swap;
Step 5: for i = 0 to a
          Print p[i] // no. of vehicles in segments
Step 6:  for I= a to n
          Print p[i]// no. of vehicles.
```

Next we are applying the Dijkstra Algorithm for finding the shortest path by which we get optimal result of the RSU Placement point or junctions

```
function Dijkstra(Graph, source):

  create vertex set Q

  for each vertex v in Graph:
      dist[v] ← INFINITY
      prev[v] ← UNDEFINED
      add v to Q

   dist[source] ← 0

   while Q is not empty:
       u ← vertex in Q with min dist[u]

       remove u from Q
    for each neighbor v of u:
           alt ← dist[u] + length(u, v)
           if alt < dist[v]:
               dist[v] ← alt
               prev[v] ← u
       return dist[], prev[]
```

RESULT AND DISCUSSION

```
#include<stdio.h>

int main (void)
{
int p[5],w[5],i,j,p1=0,n,x;
int wl=0;
printf("\n Enter the number of junctions");
scanf("%d",&n);
printf("\n Enter the number of segment you want to divide");
scanf("%d",&x);
int a=n/x;
for(i=0;i<n;i++)
{
      printf("\n Enter the name of the Junction: %d = ",i+1);
      scanf("%d",&p[i]);
      printf("\n Enter the number of vehicles: %d = ",i+1);
      scanf("%d",&w[i]);

}



for(i=0;i<a-1;i++)
{
      for(j=i+1;j<a;j++)
      {

            if(w[i]<w[j])
            {
                  p1=p[i];
                  wl=w[i];
                  p[i]=p[j];
                  w[i]=w[j];
                  p[j]=p1;
                  w[j]=wl;
            }
      }
}
for(i=a;i<n-1;i++)
{
      for(j=i+1;j<n;j++)
      {

            if(w[i]<w[j])
            {
                  p1=p[i];
                  wl=w[i];
                  p[i]=p[j];
                  w[i]=w[j];
                  p[j]=p1;
                  w[j]=wl;
            }
      }
}
printf("\n junction points where RSU can able to placed  ");
for(i=0;i<a;i++)
{
      printf(" \n %d",p[i]);
}
for(i=a;i<n;i++)
{
      printf(" \n %d",p[i]);
}
return 0;
}
```

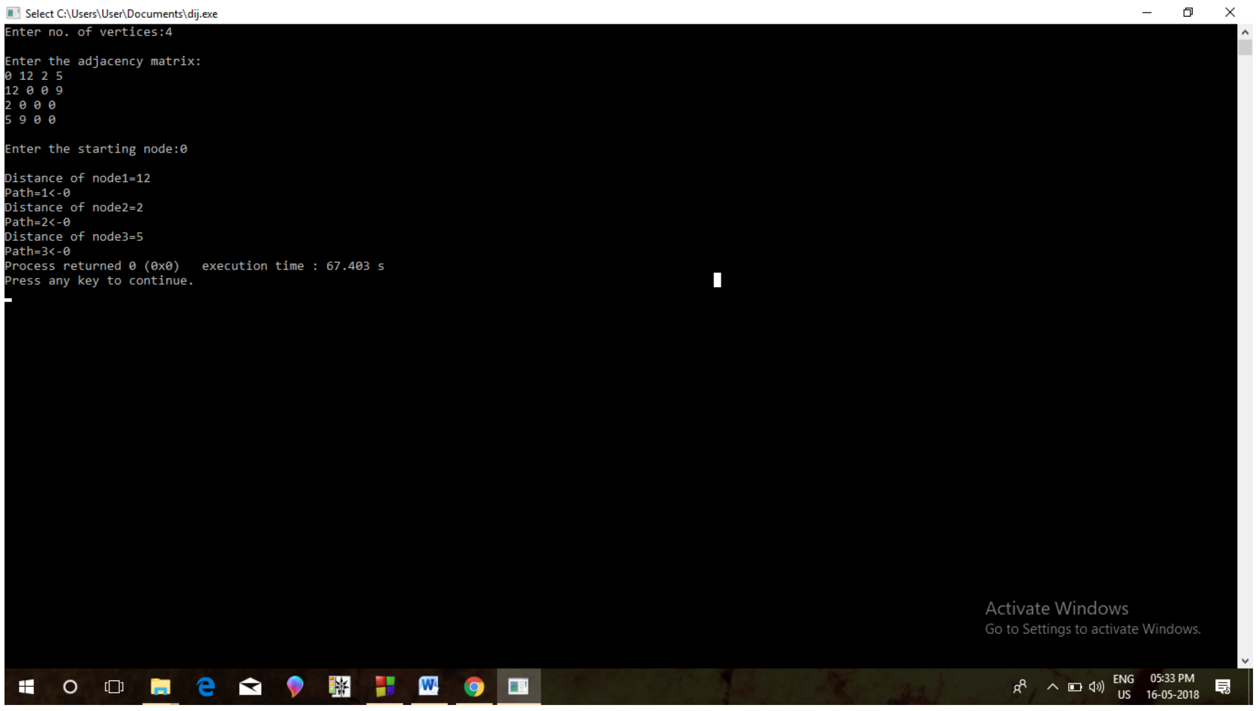


```
 Enter the number of junctions4

 Enter the number of segment you want to divide2

 Enter the name of the Junction: 1 = 1

 Enter the number of vehicles: 1 = 44

 Enter the name of the Junction: 2 = 2

 Enter the number of vehicles: 2 = 55

 Enter the name of the Junction: 3 = 3

 Enter the number of vehicles: 3 = 88

 Enter the name of the Junction: 4 = 4

 Enter the number of vehicles: 4 = 99

 junction points where RSU can able to placed   2 1 4 3
Process returned 0 (0x0)   execution time : 463.592 s
Press any key to continue.
```

Dijkstra Algorithm C Code:

```
#include<stdio.h>
#include<conio.h>
#define INFINITY 9999
#define MAX 10

void dijkstra(int G[MAX][MAX],int n,int startnode);

int main()
{
   int G[MAX][MAX],i,j,n,u;
   printf("Enter no. of vertices:");
   scanf("%d",&n);
   printf("\nEnter the adjacency matrix:\n");

   for(i=0;i<n;i++)
      for(j=0;j<n;j++)
         scanf("%d",&G[i][j]);

   printf("\nEnter the starting node:");
   scanf("%d",&u);
   dijkstra(G,n,u);
```

```
    return 0;
}

void dijkstra(int G[MAX][MAX],int n,int startnode)
{

    int cost[MAX][MAX],distance[MAX],pred[MAX];
    int visited[MAX],count,mindistance,nextnode,i,j;

    //pred[] stores the predecessor of each node
    //count gives the number of nodes seen so far
    //create the cost matrix
    for(i=0;i<n;i++)
        for(j=0;j<n;j++)
            if(G[i][j]==0)
                cost[i][j]=INFINITY;
            else
                cost[i][j]=G[i][j];

    //initialize pred[],distance[] and visited[]
    for(i=0;i<n;i++)
    {
        distance[i]=cost[startnode][i];
        pred[i]=startnode;
        visited[i]=0;
    }

    distance[startnode]=0;
    visited[startnode]=1;
    count=1;

    while(count<n-1)
    {
        mindistance=INFINITY;

        //nextnode gives the node at minimum distance
        for(i=0;i<n;i++)
            if(distance[i]<mindistance&&!visited[i])
            {
                mindistance=distance[i];
                nextnode=i;
            }

            //check if a better path exists through nextnode
            visited[nextnode]=1;
            for(i=0;i<n;i++)
                if(!visited[i])
                    if(mindistance+cost[nextnode][i]<distance[i])
                    {
                        distance[i]=mindistance+cost[nextnode][i];
                        pred[i]=nextnode;
                    }
        count++;
    }

    //print the path and distance of each node
    for(i=0;i<n;i++)
        if(i!=startnode)
        {
            printf("\nDistance of node%d=%d",i,distance[i]);
            printf("\nPath=%d",i);

            j=i;
            do
            {
                j=pred[j];
                printf("<-%d",j);
            }while(j!=startnode);
    }
}
```



```
Enter no. of vertices:4

Enter the adjacency matrix:
0 12 2 5
12 0 0 9
2 0 0 0
5 9 0 0

Enter the starting node:0

Distance of node1=12
Path=1<-0
Distance of node2=2
Path=2<-0
Distance of node3=5
Path=3<-0
Process returned 0 (0x0)   execution time : 67.403 s
Press any key to continue.
```

CONCLUSION AND FUTURE WORK

We build the code and the algo successfully.
We want to simulate the algo in simulator in future.

REFERENCES

[1] M. Seligman, K. Fall, and P. Mundur, “Storage routing for DTN congestion control,” Wirel. Commun. Mob. Comput., vol. 7, no. 10, pp. 1183–1196, Dec. 2007.

[2] lorenzo galati Giordano and L. Reggiani, Vehicular Technologies - Deployment and Applications. InTech, 2013.

[3] M. Almeida, ADVANCES IN VEHICULAR NETWORKING. .

[4] M. Maric, “An Efficient Genetic Algorithm for Solving the MultiLevel Uncapacitated Facility Location Problem.,” Comput. Informatics, vol. 29, no. 2, pp. 183–201, 2010.

[5] T. H. Cormen, C. Stein, R. L. Rivest, and C. E. Leiserson, Introduction to Algorithms, 2nd ed. McGraw-Hill Higher Education, 2001.

[6] E. S. Cavalcante, A. L. L. Aquino, G. L. Pappa, and A. A. F. Loureiro, “Roadside unit deployment for information dissemination in a VANET,” in Proceedings of the fourteenth international conference on Genetic and evolutionary computation conference companion - GECCO Companion ’12, 2012, p. 27.

[7] B. Aslam, F. Amjad, and C. C. Zou, “Optimal roadside units placement in urban areas for vehicular networks,” in 2012 IEEE Symposium on Computers and Communications (ISCC), 2012, pp. 000423–000429.

