[PLACEMENT OF RSU IN URBAN AREA]

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

BACHELOR OFTECHNOLOGY
In
INFORMATIONTECHNOLOGY

By

DEEPTANU CHATTERJEE

UNIVERSITY ROLL NO - 11700214031

DEBJIT AUDHYA

UNIVERSITY ROLL NO - 11700215094

AMLAN MAHISH

UNIVERSITY ROLL NO - 1170021502

UNDER THE SUPERVISION OF

ABANTIKA CHOUDHURY
[Assistant Professor]
[M.Tech.(Software Engineering)]

[RCC INSTITUTE OF INFORMATION TECHNOLOGY]

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

श्रमम् बिना न किमपि साध्यम्

Approval

This is to certify that the project report entitled "PLACEMENT OF RSU IN URBAN AREA" prepared under my supervision by DeeptanuChatterjee (IT/2014/079), DebjitAudhya (IT/2015/L12), AmlanMahish (IT/2015/L09), be accepted in partial fulfillment for the degree of Bachelor of Technology in Information Technology.

It is to be understood that by this approval, the undersigned does not necessarily endorse or approve any statement made, opinion expressed or conclusion drawn thereof, but approves the report only for the purpose for which it has been submitted.

……………………………………..
Dr. Abhijit Das

(H.O.D.)

(Asst. Professor)
Department of Information Technology,
RCCIIT

……………………………………………
Mrs.AbantikaChoudhury

(Asst Prof.)

Department of Information Technology,
RCCIIT

ACKNOWLEDGEMENT

I express my sincere gratitude to Mrs.AbantikaChoudhury 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 all of our department who helped me in their own way whenever needed.

Date:……………..2018

(DeeptanuChatterjee)
Reg. No.: 141170110166
Roll No.: 11700214031
B. Tech (IT) – 8th Semester,
Session-2014-2018, RCCIIT

(DebjitAudhya)
Reg. No.: 151170120016
Roll No.: 11700215094
B. Tech (IT) – 8th Semester,
Session-2014-2018, RCCIIT

(AmlanMahish)
Reg. No.:
151170120014Roll No.:
11700215092
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 titled [Put your project title here] submitted by DeeptanuChatterjee (Roll No.:11700214031 of B. Tech. (IT) 8thSemester of 2018), DebjitAudhya (Roll No.:11700215094 of B. Tech. (IT) 8thSemester of 2018),AmlanMahish (Roll No.:11700215092 of B. Tech. (IT) 8thSemester of 2018) is hereby recommended to be accepted for the partial fulfillment of the requirements forB Tech (IT) degree in MaulanaAbulKalam Azad University of Technology,West Bengal.

Name oftheExaminer	**Signature withDate**
1……………………………………….	………………………………
2………………………………………..	………………………………..
3………………………………………	………………………………
4……………………………………….	………………………………

TABLE OF CONTENTS

INTRODUCTION

In the last few years, accompanying the massive deployment of wireless technologies and the growing number of wireless products on motorized vehicles including remote keyless entry devices, personal digital assistants (PDAs), laptops, and mobile telephones, automotive industries have opened a wide variety of possibilities for both drivers and their passengers. Vehicular Ad hoc Networks (VANETs) have attracted a lot of attention in research community because of their varied value added services, namely vehicle safety, automated toll payment, traffic management, enhanced navigation, location-based service for finding the closest fuel station, travel lodge or restaurant and simply access to the Internet .

Vanet- Vehicular Ad-Hoc Network is the network in which communication has been done in between road side units to cars, car to car in a short range of 100 to 300 m. Existing authentication protocols to secure vehicular ad hoc networks (VANETs) raise challenges such as certificate distribution and revocation, avoidance of computation and communication bottlenecks, and reduction of the strong reliance on tamper-proof devices. In a VANET, vehicles will rely on the integrity of received data for deciding when to present alerts to drivers. Further in the future, this data may be used as the basis of control decisions for autonomous vehicles. If this information is corrupted, vehicles may present unnecessary or erroneous warnings to their drivers, and the results of control decisions based on this information could be even more disastrous. Information can be corrupted by two different mechanisms: malice and malfunction. Similarly, vehicles have two defense mechanisms: an internal filter and external reputation information.

streaming over end to end Internet Protocol. If the Internet Protocol (IP) multimedia sub-system movement achieves what it going to do, nothing of this possibly will matter. The former defense mechanism can consist of filters based on physical laws (e.g., maximum braking deceleration, maximum speed, physical space constraints) [2]. The latter defense mechanism can consist of reports from other vehicles or entities on the validity or trustworthiness of data originating from certain .[1] In this paper, we will concern ourselves with the latter defense mechanism. Information received from corrupted nodes should be disregarded or not trusted by legitimate vehicles, otherwise, a malicious vehicle could, for example, obtain a less congested route for itself by overstating the number of vehicles on its desired roadway. As a Second example, a corrupted node could trigger erroneous driver warnings to be displayed in other vehicles by falsifying its position information. IEEE 1609.2, the trial-use standard concerning security services for vehicular environments, stipulates that vehicles will be authenticated using certificates issued by a Certificate Authority (CA) in a Public Key Infrastructure (PKI) setup [3]. Illegitimate vehicles should have these certificates revoked, and the identity of the revoked certificates (although ideally not the identity of the associated driver) should be published and distributed to legitimate vehicles. Whatever mechanism that is used for distributing this revocation information should distribute the info information securely, quickly, and broadly in order to limit the amount of damage illegitimate vehicles can do. First we discuss the general architecture and security architecture of vanet. Next our paper addresses the analytical evaluation of different research paper in VANET. Than we compare different popular.

PROBLEM DEFINATION

As we know that deployment of many RSU s is very costly. If we deploy many RSU s then the total cost of the project will increase to huge amount. So wecant work only with the RSU and Vehicle connection ,We also have to work with the vehicle to vehicle connection. But there is a problem arise that if the vehicles quantity in road is less then the connection will not be established properly. So we are developing and adding an algorithm to the project which will help to deploy minimum RSU s to ensure a established connection like R2V , V2 V, V2R..
Previous studies on VANET routing focused more on single ad hoc routing method (e.g. most researchers focused on traditional ad hoc topology based routing, while some other focused on position based ad hoc routing method in VANET). The selection of routing method heavily depends on the nature of the network. Thus single ad hoc routing method is not sufficient enough in meeting all the different types of ad hoc networks. In this study we focus on different ad hoc routing methods and figure out which recent advancement had been made to provide „in time‟ and scalable routing in order to avoid any critical situation on roads. Furthermore, most researchers focused on single environment of VANET i.e. either on highway or in city to evaluate the performance of different routing protocols. Therefore in our study we focus on both environments i.e. city and highway for the performance evaluation of different routing protocols. Moreover, the performance of different routing protocols had not been well measured since each researcher used different simulator and performance metrics for performance evaluation. Due to aforementioned problems there is continuous need to study various ad hoc routing methods in order to select appropriate method for different environments of VANET.

LITERATURE STUDY

1. Celimuge Wu, YushengJi, Fuqiang Liu, Satoshi Ohzahata& Toshihiko Kato

proposed routing protocol which is able to learn the best transmission parameters by interacting with the environment. The protocol takes into account multiple metrics specifically data transmission rate, vehicle movement and route length. They used both real-world experiments and computer simulations to evaluate the proposed protocol. The protocol employs a Q-Learning-based approach to estimate transmission rate from the hello packet reception ratio. For the route selection, the protocol uses a fuzzy logic-based algorithm to evaluate the direct link and uses a Q-Learning algorithm to learn the best end-to-end route. The proposed protocol has been implemented with Ubuntu 12.04 and then evaluated using a real vehicular ad hoc network. Q-Learning is a form of reinforcement learning algorithm that works by estimating the values of state action pairs without requiring a model of its environment. Q-Learning adjusts behavior through trial-and-error interactions with a dynamic environment.

2. SeonYeong Han, Dongman Lee

conveys that in mobile ad-hoc networks, local link connectivity information is extremely important for route establishment and maintenance. However, unnecessary Hello messaging can drain batteries while mobile devices are not in use. This paper proposes an adaptive Hello messaging scheme to suppress unnecessary Hello messages without reducing detects ability of broken links. Simulation results show that the proposed scheme reduces energy consumption and network overhead without any explicit difference in throughput. In this paper, they proposed an adaptive Hello interval to reduce battery drain through practical suppression of unnecessary Hello messaging.

3. Christian Lochert, Hannes Hartenstein et al.

analyzed a position based routing protocol which can work under high mobility of nodes. They utilize the navigation system for their work. With the help of simulation they compare their approach with the no position based approach by using the highly realistic mobility of the vehicles. They concluded their approach as “geographic source routing” (GSR) which combines both position and topology approach. They found that GSR performs much better than the topology based approaches that are DSR and AODV on the basis of delivery rate and latency

4. F. Dotzer, L. Fischer, and P. Magiera [19] They proposed that VARS (Vehicle Ad-Hoc Network Reputation System) is a reputation-based system which uses modules for direct and indirect reputation handling, opinion generation and confidence decision (message handling) and situation recognition. They concluded that VARS defines three areas: the event area within which an event can be recognized, the decision area where the trustworthiness of event messages have to be decided upon and the distribution area which specifies how far those messages are distributed.

PROBLEM DISCUSSION

Our main focus is placement of RSU in urban area by which we can reduce the cost and reduced the delay of the communication. RSU is the main component of the VANET architecture we have to place the RSU in an accurate place by where we can get the total sufficient throughput from the RSU. Then the after we can able to create the VANET architecture in urban area.

Advantages of Project

- It is very easy to implement and dense a large network with the help of simulator.
- The implementation cost is less.
- This is less time consuming.
- 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.

Limitation of Project

- Some real time conditions are not possible to implement.
- Coding is complex and lengthy.
- During simulation the real time frequent topology changes are difficult to achieve.

SRS(Software Requirement Specification)

RSU Placement is the starting job of creating VANETarchitecture
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 DijsktraAlgorithm 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

HardwareNeeded :

- 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

We are working in an algorithm of placement of RSU
At the first we have to take the junctions and numbers of RSU want to place , after that check the Vehicle density and the we have to put the values of Vehicles passing from the junctions and name the junctions

Input: Name the junctions
Enter the numbers of vehicles
Enter the no. of RSU want to place
Output: Give the junction name where maximum no. of car passing from.

```
1.  Initialize the array of junctions, no. of vehicle and also no. of RSU want to place.
2.  for i= 0 to n
        Enter the name of the junctions and no. vehicles passing from the junctions
3. for i= 0 to n-1
        For j=0 to n
                v[i]<v[j]
                {
                        Swap the junctions;
                }
4.for i= 0 to n
        Print the junction name;
```

We are implement this algorithm in to our project and see that how this algo working in RSU placement in urban area where the traffic density is higher.

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

```
1  functionDijkstra(Graph, source):
2
3      create vertex set Q
4
5      for each vertex v in Graph:
6          dist[v] ← INFINITY
7          prev[v] ← UNDEFINED
8          add v to Q
9
10      dist[source] ← 0
11
12      whileQ is not empty:
13          u ← vertex in Q with min dist[u]
14
15          remove u from Q
16       for eachneighborv of u:
17alt ← dist[u] + length(u, v)
18ifalt<dist[v]:
19dist[v] ← alt
20prev[v] ← u
21         returndist[], prev[]
```

If we are only interested in a shortest path between vertices *source* and *target*, we can terminate the search after line 15 if *u* = *target*. Now we can read the shortest path from *source* to *target* by reverse iteration:

```
S ← empty sequence
u ← target
whileprev[u] is defined:
    insert u at the beginning of S
    u ← prev[u]
insertu at the beginning of S
```

RESULT AND DISCUSSION

First we have to find the junctions where maximum no. of cars are passing from, next we have to place those junctions to find the shortest path where we can place the RSU and we will get the total range of the RSU.

First algorithm C Code:-

```
#include<stdio.h>

int main (void)
{
int p[5],w[5],i,j,p1=0,rsu,n;
intwl=0;
printf("\n Enter the no. of junctions");
        scanf("%d",&n);
        printf("\n Enter the maximum no. RSU want to place");
        scanf("%d",&rsu);
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<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<rsu;i++)
{
        printf(" %d",p[i]);
}
return 0;
}
```

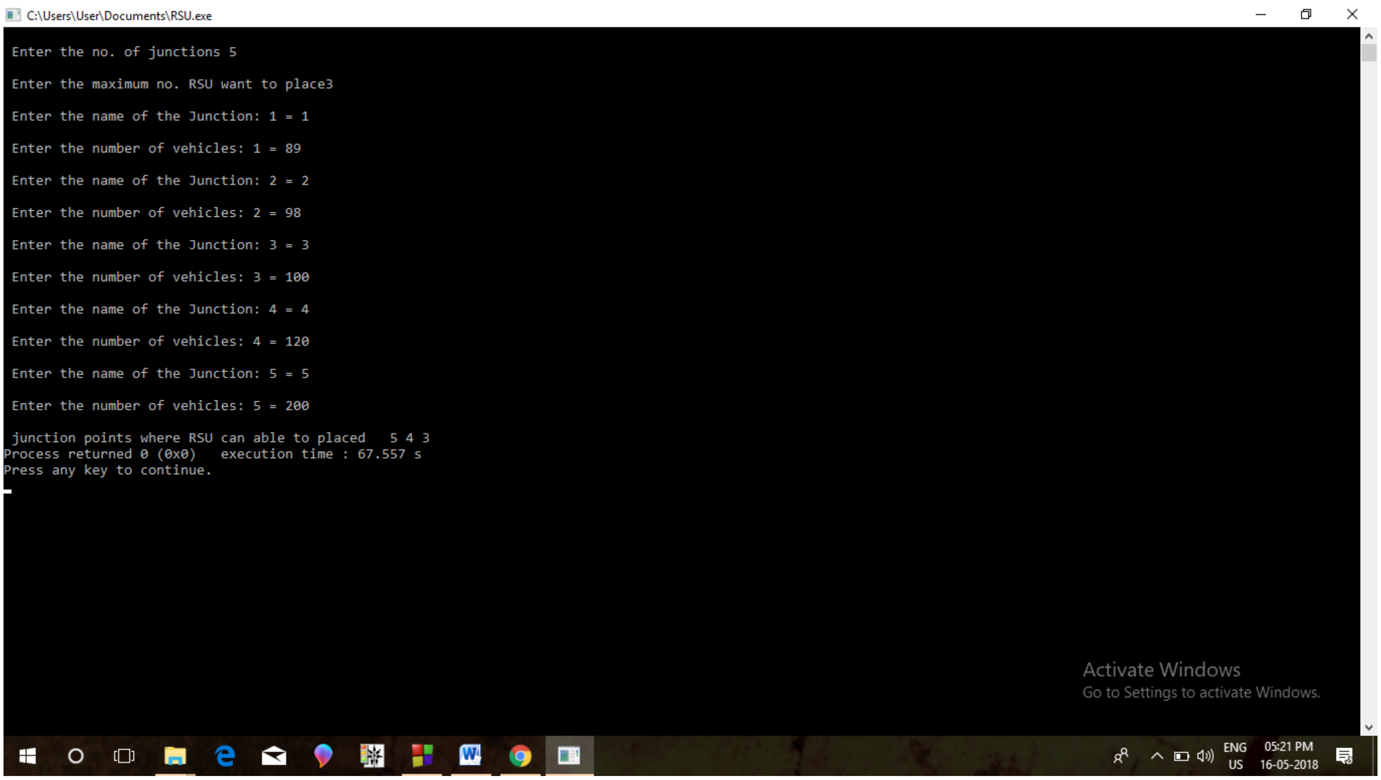

Fig 1:- Output of first Algorithm

Dijkstra Algorithm C Code:

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

voiddijkstra(int G[MAX][MAX],intn,intstartnode);

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;
}

voiddijkstra(int G[MAX][MAX],intn,intstartnode)
{

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);
   }
}
```

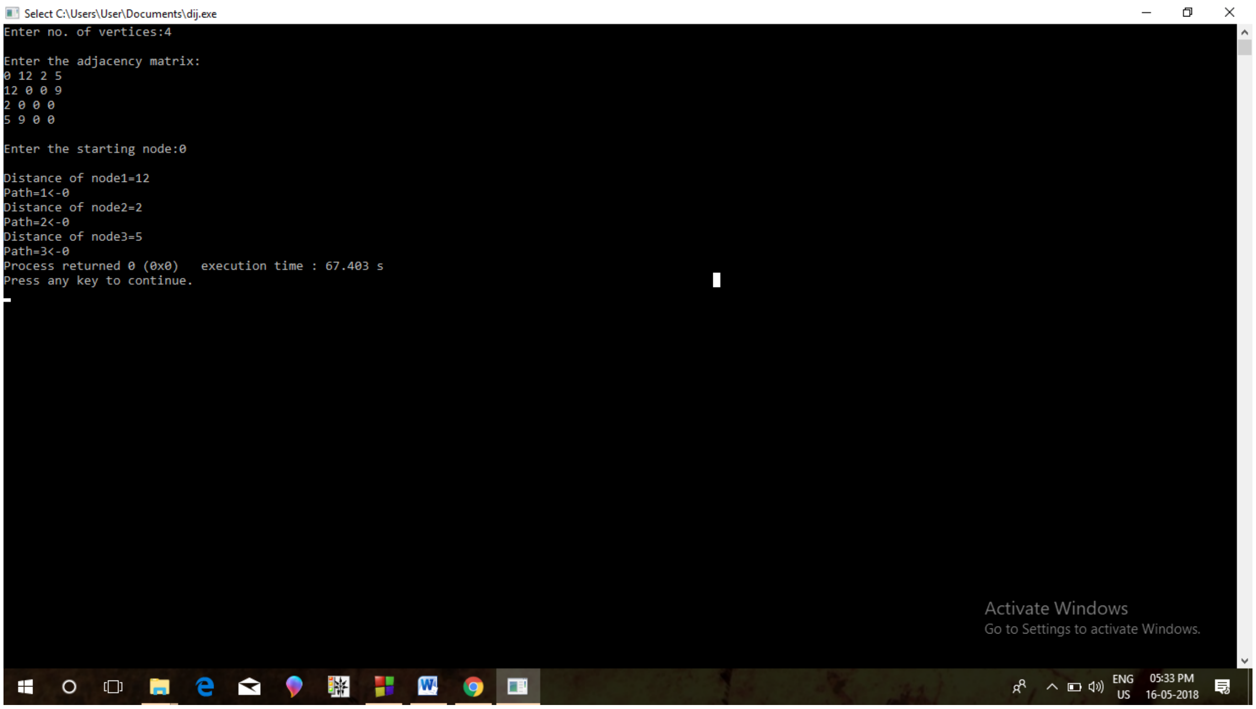

Fig 2:- Output of Dijkstraalgo.

CONCLUSION

We successfully build the code and the algorithms by which we can able to reduce the delay of the RSU communication and also it is cost efficient.

In future we want to run it in the simulator by which we can get a appropriate result of this VANET architechture.

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] lorenzogalati 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 MultiLevelUncapacitated 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.

[8] Y. Sun, X. Lin, R. Lu, X. Shen, and J. Su, “Roadside Units Deployment for Efficient Short-Time Certificate Updating in VANETs,” in Communications (ICC), 2010 IEEE International Conference on, 2010, no. January 2014, pp. 1–5.

[9] Y. Liu, J. Niu, J. Ma, and W. Wang, “File downloading oriented Roadside Units deployment for vehicular networks,” J. Syst. Archit., vol. 59, no. 10, pp. 938–946, Nov. 2013.

[10] O. Trullols, J. M. Barcelo-Ordinas, M. Fiore, C. Casetti, and C.-F.Chiasserini, “A Max Coverage Formulation for Information Dissemination in Vehicular Networks,” in 2009 IEEE International Conference on Wireless and Mobile Computing, Networking and Communications, 2009, pp. 154–160.

[11] İ. K. Altınel, N. Aras, E. Güney, and C. Ersoy, “Binary integer programming formulation and heuristics for differentiated coverage in heterogeneous sensor networks,” Comput. Networks, vol. 52, no. 12, pp. 2419–2431, 2008.

[12] Y. Xiong, J. Ma, W. Wang, and J. Niu, “Optimal Roadside Gateway Deployment for VANETs,” Prz. ELEKTROTECHNICZNY, no. 7, pp. 273–276, 2012.