ISSN 2347 - 3983
Volume 9. No.11, November 2021
International Journal of Emerging Trends in Engineering Research
Available Online at http://www.warse.org/IJETER/static/pdf/file/ijeter079112021.pdf
https://doi.org/10.30534/ijeter/2021/079112021

Dynamic Routing of Emergency Vehicles with Traffic Management System and Implementation on CupCarbon for Smart Cities

Ahmet Mirza Yildirim[1], Adnan Kavak[2]
[1]Kocaeli Univ.,Computer Eng. Dept,41380 Izmit,Turkey,185112018@kocaeli.edu.tr
[2]KocaeliUniversity, Computer Eng. Dept, 41380 İzmit, Turkey, akavak@kocaeli.edu.tr

Received Date : October 06, 2021 Accepted Date : October 27, 2021 Published Date : November 07, 2021

ABSTRACT

The traffic congestion is one of the major problems in crowded cities, which causes people to spend hours on the road. In traffic congestion situations, finding alternate routes for emergency vehicles, which provides shortest travel time to nearby hospital is critically life-saving issue. In this paper, we propose a traffic management system and an algorithm for routing of an emergency vehicle. The algorithm uses distance between source and destination, maximum vehicle count, maximum speed, average speed, traffic light conditions on the roads, which are assumed to support vehicle-to-infrastructure (V2I) communication in 5G IoT network. Simulations are performed on CupCarbon IoT simulator platform for various test scenarios. The performance of the proposed emergency vehicle routing algorithm is compared against well known Link State algorithm. And, the results demonstrate the effectiveness of the proposed method.

Key words : Cupcarbon, Routing, Smart City, Traffic Congestion, Traffic Management, 5G

1. INTRODUCTION

Road transportation is the mostly used type of transportation in the world. It is also the most dangerous of all transportation methods. According to the World Health Organization (WHO), 1.35 million people die in traffic accidents every year. According to the Turkish Statistical Institute, there were 1,168,144 traffic accidents in our country in 2019.1.896 of these accidents were traffic accidents by death and a total of 5,473 citizens lost their lives. Of these 5,473 citizens, 2,949 died not at the scene of the accident, but after the accident. It is a fact that hundreds of thousands of people around the world can survive if the necessary medical intervention can be done quickly. The biggest reason that prevents this rapid intervention is traffic congestion. Due to traffic congestion, people spend many hours on the roads every day. This is perhaps the greatest impact of improper urbanization on human life.

In addition, new vehicles are coming to the roads every day and these problems are deepening. Today, traffic shortages and the success of traffic management are the biggest factors that make big cities classify as livable. Considering all these losses, the most important part of the concept of smart city is the management and routing of traffic under the heading Smart Traffic Management Systems. With the developing technology, some solutions to one of the most important problems of this modern age are tried to be developed. IoT and 5G technologies, which are the most trending topics of recent times, are technologies that can help solve this problem a little bit. These emerging technologies and communication protocols provide the infrastructure needed to create more performance and flexible Intelligent Traffic Management Systems.

Although there are various researches and solutions especially on routing in the literature. These solutions are usually based on distance between source and destinations or instantaneous speeds of vehicles. In addition, these methods cover certain and very limited areas. In this work, we specifically focus on the effect of traffic light with this regard, since the red light situation that affects average speed of vehicles isone of the key factors causing delays during routing process. In existing traffic monitoring systems such as in Istanbul, travel time estimation is given based on the average speed values on that road.

In this paper, we aim to determine the fastest route by communicating with the infrastructure (V2I) until the vehicle arrives the destination, taking into account the latency according to the current condition and characteristics of the road. We use CupCarbon for simulation[1],[2].CupCarbon is a smart city, WSN and IoT simulator [3]. It offers two simulation environments. On the one hand, it enables the design of scenarios with mobility and generation of natural events and on the other, the simulation of discrete events in WSNs. The network can be designed thanks to its intuitive graphical interface making use of the OpenStreetMap (OSM) where the sensors can be placed directly. In addition, each sensor can be individually configured by its command-line making use of a language specially designed for CupCarbon called SenScript [4].

2. RELATED WORKS

In literature, there exists methods for management of traffic systems. Faisal et al. have tested two scenarios for the most effective management of accident situation in the Jordanian city due to the increasing number of vehicles and traffic density [5]. In the first scenario, the ambulance notifies a control center of the accident situation after the accident. The control center receives status information from the hospitals closest to the crash site and calculates the length of the existing routes between the accident location. It sends the shortest route information to the ambulance and the accident information to the hospital. In the second scenario, it sends the most appropriate route to the ambulance, not only according to the length of the roads but also according to the average speed data of the roads. In [6], authors have proposed traffic management system for prioritization of emergency vehicles according to the type of a vehicle (ambulance, police car, fire brigade and normal cars), the priority of the incident (according to the vehicle type), traffic light status, the number of vehicles available, and the hacking status of the traffic signaling system. Emergency vehicle planning will be done according to information level at road side unit (RSU). At each intersection, the RSU will first check the accuracy of the information packets and if the number is above the predetermined threshold, that RSU will be considered hacked and that RSU's information will be shifted for processing in another RSU. In [7],travel time of an ambulance to a hospital has been tried to be minimized by adjusting the speed of the vehicles around it and the traffic lights according to it.

3. SYSTEM COMPONENTS AND OPERATION

3.1. System Components

This section describes the system components and their operation used in our modeling for emergency vehicle routing.

Figure 1: The Snapshot of a Simulation Model on CupCarbon

There are basically three main components in our model. The orange box represents an emergency vehicle, the yellow box represents the road side units (RSU) along the path of the vehicle, the green box denotes the central management system (CTM), and the red box shows the target node. The RSUs are located at a distance where they can send data to the CTM. In addition, the CTM is positioned to exchange data with all system components. There are three different communication protocols, i.e. ZigBee, Wi-Fi, and Lora, defined for operation of sensor nodes in CupCarbon [8].Which protocol to use depends on the system to be used. Sensors are represented in three different colors according to their communication types as shown in Figure 2.

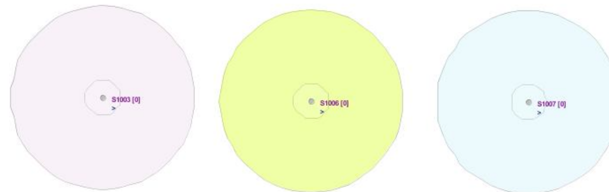

Figure 2: ZigBee, WiFi and Lora sensor images, respectively, used in CupCarbon.

RSUs are assumed to keep the information such as path length, maximum vehicle capacity, maximum speed capacity, instantaneous average speed information, instant number of vehicles, and traffic light status of the road they are responsible for. The emergency vehicle sends the request message for the shortest route to the destination to RSUs along its path. The code piece that implements vehicle-to-RSU communication is given in Figure 3.

```
X:loop
        send iamhere 0 2
        wait 300
        read cmd
        cbuffer
        length lng $cmd
        if($lng==0)
                goto X
        end
        rdata $cmd msg nodeIndex
        if($msg==okey)
                tget isControlled nodes $nodeIndex 0
                if($isControlled==0)
                        delay 300
                        send routeReq 0 2
                        delay 100
                        tset 1 nodes $nodeIndex 0
                    end
        else
                        plus indexfrom $indexfrom 1
                        plus indexto $indexfrom 1
                        charat from $cmd 0
                        charat to $cmd 1
                        function newpath cusconc $from,$to
                        route $newpath
        end
```

Figure 3: SenScriptCode Snippet That PerformsVehicle-to-RSUCommunication in CupCarbon.

The RSU adds the vehicle information to this request and sends it to the CTM. Then, CTM broadcasts this request to all RSUs within its coverage. The proposed algorithm is assumed to run at CTM, and it sends the most appropriate route information to the requesting vehicle. If the vehicle is not at the destination node when it arrives at the next node, the same process starts again. The code piece that implements CTM function is given in Figure 4.

```
loop
        wait
          read v
          length lng $v
          if($lng>0)
            plus index $index 1
            rdata $v Name Vmax Cmax Vort Cort I X Passspeed
            tset $Name tb $index 0
            tset $Vmax tb $index 1
            tset $Cmax tb $index 2
            tset $Vort tb $index 3
            tset $Cort tb $index 4
            tset $I tb $index 5
            tset $X tb $index 6
            tset $Passspeed tb $index 7
            if($index==22)
                set str \
                for i 0 23
                   tget name tb $i 0
                   tget vmax tb $i 1
                   tget cmax tb $i 2
                   tget vort tb $i 3
                   tget c tb $i 4
                   tget light tb $i 5
                   tget x tb $i 6
                   tget passpeed tb $i 7
                   function dt cusconc -,$name,#,$vmax,#,$cmax,#,$vort,#,$c,#,$light,#,$x,#,$passpeed
                   function str cusconc $str,$dt
                   delay 100
                end
                function path getpath $str,3,$nodeIndex
                send $path 1001
                goto B
          end
        end
```

Figure 4: SenScriptCode Snippet for CTM in CupCarbon.

3.2.Proposed Emergency Vehicle Routing Algorithm

Flowchart of our proposed emergency vehicle routing algorithm is given in Figure 5. Basic idea of this method is to find shortest travel time based on various parameters obtained via RSUs at each intersection where various alternative paths are available to follow for an emergency vehicle. The parameters used in this algorithm are calculated as given in Equations (1) through (4) below, and they are summarized in Table 1.

$$\boldsymbol{T(V_{max}, C_{max}, C, V_{ort}, I, I_t, d)} = \begin{cases} \dfrac{d}{V_{ort}} & , \quad I = 0 \\ G(V_{max}, C_{max}, C, I_t, d), & I = 1 \end{cases}$$

(1)

$$\boldsymbol{G(V_{max}, C_{max}, C, I_t, d)} = \begin{cases} \dfrac{d}{H(V_{max}, C_{max}, C)} & , \quad I_s(V_{max}, C_{max}, C, I_t) < 90 \\ \dfrac{d}{H(V_{max}, C_{max}, C)} + \left(I_t - \dfrac{d}{H(V_{max}, C_{max}, C)}\right), & I_s(V_{max}, C_{max}, C, I_t) \geq 90 \end{cases}$$

(2)

$$\boldsymbol{H(V_{max}, C_{max}, C)} = V_{max} - \left(V_{max} \times \frac{|C_{max} - C_{ort}|}{C_{max}}\right)$$

(3)

$$\boldsymbol{I_s(V_{max}, C_{max}, C, I_t, d)} = \frac{I_t \times 100}{H(V_{max}, C_{max}, C)/d}$$

(4)

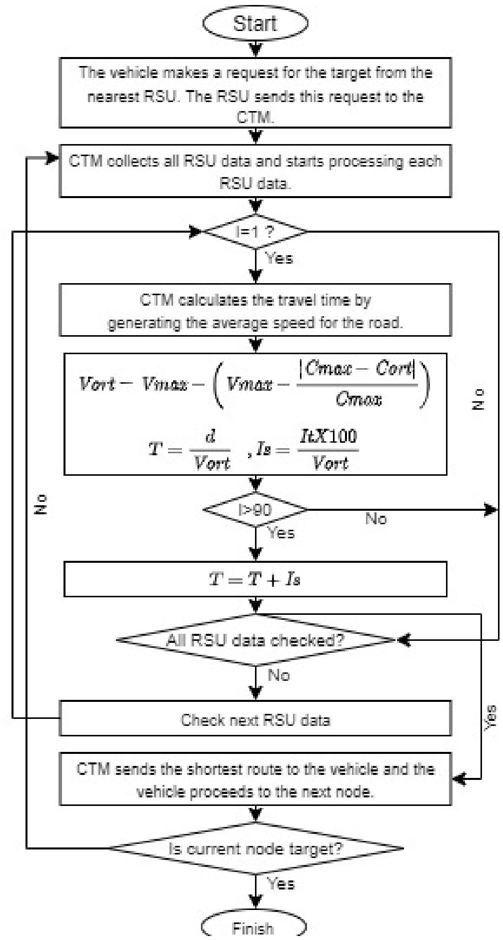

Figure 5: Flowchart of the Proposed Traffic management system for emergency vehicle routing.

3.3.Data Generation in CupCarbon

In order to generate data while simulating in Cupcarbon, some instantly changing data of the roads and some unchanging data depending on the physical conditions of the road must be known. This information is generated and stored by RSUs, when a request is made by the CTM as shown in Figure 6.

Table 1: Parameters Used in the Algorithms

Symbol	Description
T	main cost function
G	gives travel time when $I = 1$
F	gives V_{ort} when $I = 0$
H	gives V_{ort} when $I = 1$
I_s	I_t is what percent of travel time
V_{max}	max speed(km/h)
V_{ort}	instant average speed
C_{max}	max vehicle count
C_{rate}	C is what percent of C_{max}
C	instant count of vehicles
I	traffic light(0=green;1=red)
I_t	traffic light time(s)
d	road length(km)
Rnd	gives random number between given params

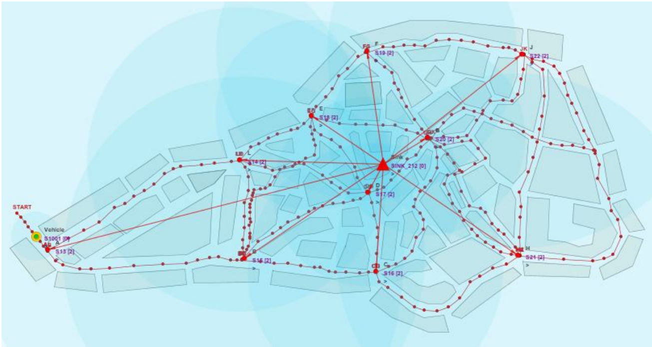

Figure 6: CTM requesting data from All RSUs

The piece of a code on CupCarbon for generating data at the RSU named A is given in Figure 7. Firstly the light status, i.e, red or green is generated using Binomial distribution. Then, if the light status is red light, i.e., $I = 1$,the light duration I_t is randomly generated between 10-210 seconds. Then, instantaneous number of vehicles$C$ is generated using Poisson distribution according to maximum number of vehicles C_{max}. Then,average speed of vehicles V_{ort} is generated as in Equation 6 using V_{max},C_{max}, C values.

```
if($cmd==ReqData)
        function light getlight 0,AB
        function count getcount 143,AB
        function passspeed getpassspeed 120
        function speed getspeed 156,$count,143,AB
        data dt AB 156 143 $speed $count $light 3 $passspeed
        send $dt 212
        delay 150

        function light getlight 0,AL
        function count getcount 169,AL
        function passspeed getpassspeed 156
        function speed getspeed 156,$count,169,AL
        data dt2 AL 156 169 $speed $count $light 4 $passspeed
        send $dt2 212
        delay 150
end
```

Figure 7: SenScriptCode that Generates Data for RSU Named A (for AB and AL routes) and sends it to CTM

$$C_{rate} = \frac{C \times 100}{C_{max}} \quad (5)$$

$$F(x) = \begin{cases} Rnd\left(\left(\frac{V_{max} \times C_{rate}}{100}\right), V_{max}\right), & C_{rate} \leq 100 \\ Rnd\left(0, \left(V_{max} - \frac{(V_{max} \times (C_{rate} - 100))}{100}\right)\right), & C_{rate} > 100 \end{cases}$$

(6)

The Rnd function generates a random number between the given parameters. With the Rndfunction, an average speed value of the previous times is generated between 30% and 80% of the V_{max}value. This value will be used as a general average speed value, not instantaneous, for this path. RSUs send this data they produce to the CTM. Then the CTM determines the shortest path from the current node to the target node and sends the next node to the vehicle, and the vehicle follows the path along that node. If the new node is not the target node, this process repeats until the vehicle arrives the target node.

4. TEST SCENARIOS AND RESULTS

Various test scenarios are implemented on CupCarbon to assess the performance of our proposed algorithm. The V_{max} , C_{max} and d values generated for each path in simulation model in Figure 1 are generated as shown in Table 2.

Table 2: Default Data of the Roads Used in the Simulation Model.

Path	V_{max}(km/h)	C_{max}	d (km)
AB	50	15	3
AL	40	40	1
BC	120	200	3
BE	90	150	2
BD	70	50	4
CH	120	160	5
CG	100	140	3
CD	40	30	2
DG	30	15	1
DE	60	50	2
EF	90	100	3
EG	50	60	3
FG	90	120	2
FJ	70	50	4
GJ	120	300	3
GH	80	90	2
GK	120	130	4
HK	70	60	3
HJ	90	200	4
HG	40	30	2
JK	70	30	2
LD	120	150	3
LE	80	50	1
LB	100	40	2

In the first scenario, the simulation was run with the default values shown in Table 2. The results were obtained and complementary CDF statistics of travel time is plotted as shown in Figure 8. If we consider 50 percentile values, our proposed algorithm provides travel time of 3.48 minutes, and that of the traditional link state algorithm is found to be 4.01 minutes. If our proposed algorithm is used, the vehicle arrives at the destination 0.53 minutes earlier.

Each road has an average number of vehicles that it can handle without creating traffic (C_{max}). As we approach this number, traffic and delays begin to occur on the road. Likewise, roads have a legally limited speed value (V_{max}). In scenario 2, we increased the values in Table 2 by 30% and observed the effect on the results. If we consider 50 percentile values in Figure 9, our algorithm has travel time of 2.40 minutes, and that of the traditional link state algorithm is found to be 3.55 minutes. If our proposed algorithm is used, the vehicle arrives at the destination 1.15 minutes earlier in this scenario.

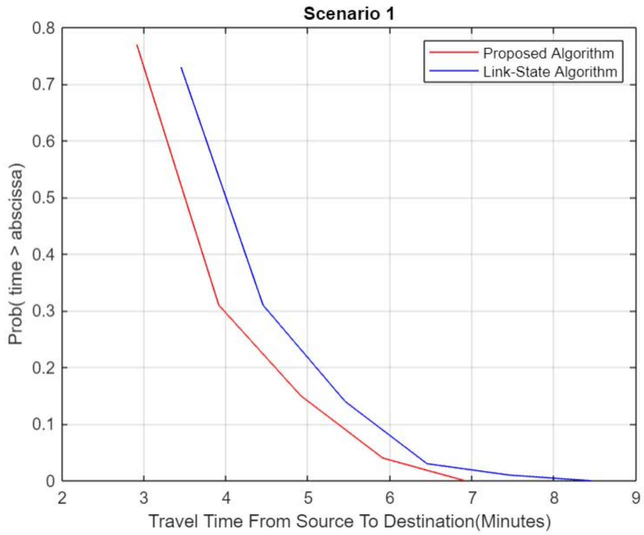

Figure 8: Complementary CDF of travel times for Scenario 1

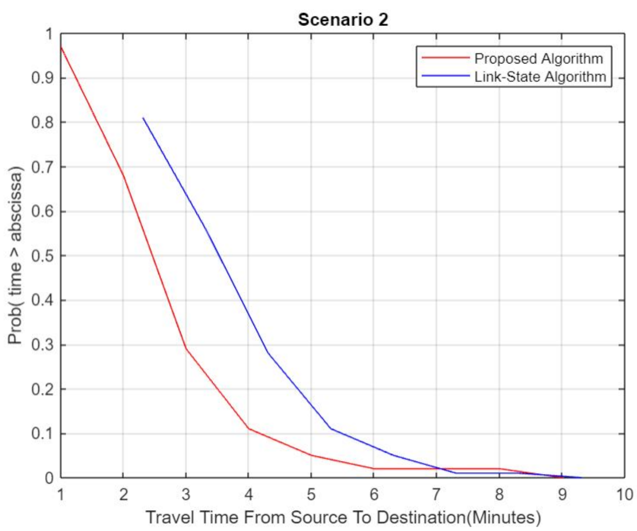

Figure 9: Complementary CDF of travel times for Scenario 2

In order to calculate the travel time when the red light is on, the average speed value is accepted as the value just before the light turns into red or the average value for all times. In our system, this value is calledV_{pass}. This value is calculated by generating a random number in the range of 10%-80% of the V_{max} value in Scenario 1. The results in scenario 3 were obtained and plotted in Figure 10 by changing the range of this value to 10%-50%.Under this case, when 50 percentile of travel times is considered, the vehicle arrives at the destination 0.91 minutes earlier when our algorithm is used.

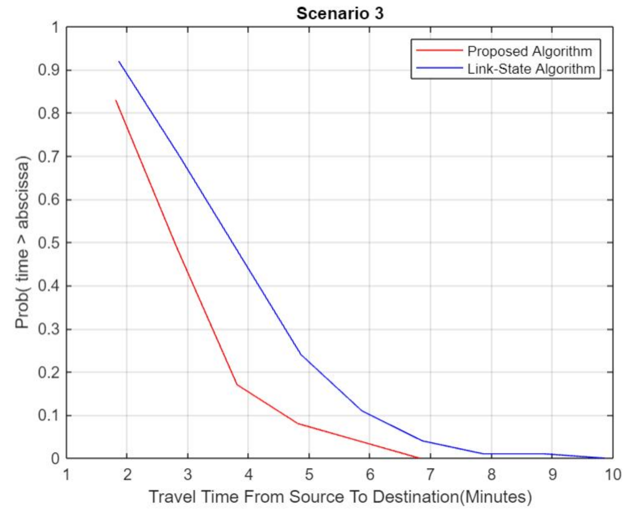

Figure 10: Complementary CDF of travel times for Scenario 3

5. CONCLUSION

In this article, we have proposed a routing algorithm for an emergency vehicle, which works in intelligent traffic management system of a smart city.The algorithms considers various factors and parameters to estimate the shortest travel time for the vehicle. This algorithm was assumed to run at CTM in traffic management system. RSUs are assumed to be deployed along the vehicle alternative paths, and they interact with CTM and vehicles to provide necessary traffic information to be used in the algorithm. The performance of our algorithm was tested under various traffic scenarios and compared with traditional link state algorithm.

REFERENCES

1. **CupCarbon IoT Simulator**, *http://cupcarbon.com.*
2. M. Akkurt and K. Küçük.**Simulation of Smart City Applications Based on IoT Technologies with CupCarbon,***In Proc. 3rd International Conference on Computer Science and Engineering*, 2018, pp. 179-184.
3. Bounceur, L. Clavier, P.Combeau, O. Marc, R. Vauzelle, A. Masserann,J. Soler, R. Euler, T. Alwajeeh, V. Devendra, U. Noreen, E. Soret, M. Lounis. **CupCarbon: A new platform for the design, simulation and 2d/3d visualization of radio propagation and interferences in IoT networks,***In Proc. 15th IEEE Annual Consumer Communications & Networking Conference (CCNC*), Las Vegas, 2018, pp. 1–4.
4. Cristina Lopez-Pavon, Sandra Sendra, Juan F. Valenzuela-Valdes.**Evaluation of CupCarbon Network Simulator for Wireless Sensor Networks,***Network Protocols and Algorithms,*vol. 10, no. 2, pp.1-27, 2018.

5. Faisal Alzyoud, NesreenAlsharman, Thamer Al-Rousan, Yahya Alsalah, Sharman Assistant. **Smart Accident Management in Jordan using Cup Carbon Simulation,***European Journal of Scientific Research*, vol.152, no.2, pp. 128-135, 2019.
6. L. Sumi, V. Ranga. **Intelligent TrafficManagement System for Prioritizing Emergency Vehicles in a Smart City,***International Journal of Engineering (IJE), IJE Trans. B: Applications,*vol. 31, no. 2,278-283, February 2018.
7. K. Rubini, M.Vidya, S.R. Yeshaswini, A. Gowthami. **Automatic Ambulance Detection and Intimation Using RSSI,** *International Journal of Emergining Technologies in Engineering Research*, vol. 7, no. 3, pp. 40-43, March 2019.
8. Szymon Rzadkosz. ***A Tale of Five Protocols,***San Francisco, 2015, ch 8.