JOURNAL OF TRANSPORTATION
SYSTEMS ENGINEERING AND INFORMATION TECHNOLOGY
Volume 14, Issue 5, October 2014
Online English edition of the Chinese language journal

Available online at www.sciencedirect.com
ScienceDirect

Cite this article as: J Transpn Sys Eng & IT, 2014, 14(5), 23−36.

RESEARCH PAPER

Transportation Routing in Urban Environments Using Updated Traffic Information Provided through Vehicular Communications

Yousefi Saleh[*1], Abbasi Tofigh[2], Anvari Zahra[1]

1 Computer Department, Faculty of Engineering, Urmia University,Urmia, Iran
2 Computer Department, Azad university-Shabestar branch, Shabestar, Iran

Abstract: Finding optimal routes for vehicles to reach their destinations is deemed to be a significant challenge and that is an objective in Intelligent Transportation Systems (ITS). In this paper, we propose a mechanism for vehicle routing based on the availability of updated traffic information. The proposed mechanism includes two phases. In the first phase, we propose a TIS (Traffic Information System) which collects the updated traffic status and stores it in the traffic information center. The proposed TIS approach consists of wired or wireless infrastructure and ad hoc inter-vehicle communications. This information will be used for vehicle navigation when a vehicle intends to run some errands. In the second phase, through exercising the available updated traffic information; we propose two different Dijkstra-based vehicle route suggestion algorithms named one-step vehicle path suggestion and step-by-step vehicle path suggestion. In the former, the algorithm is invoked only once at the beginning of the trip in order to obtain the best route towards the destination. The obtained route is then used by the vehicle throughout the journey. Nevertheless, in step-by-step path suggestion algorithm the suggested route toward the destination is being updated and refreshed at each intersection. The proposed step-by-step algorithm is further enhanced by two novel methods for avoiding loop creation. Results of the extensive simulation study using NCTUns 6.0 network simulator shows that both of the routing algorithms use the updated traffic information while the step-by-step algorithm outperforms the one-step path suggestion algorithm.

Key words: intelligent transportation systems (ITS), VANETs; WiMAX networks; road-side unit (RSU); directional antenna; dynamic vehicle routing; vehicles navigation; traffic information system

1 Introduction

The soaring trend in producing vehicles and the necessity for utilizing the personal cars as well as public vehicles in modern life, have made people reluctantly spend significant amount of time while commuting. Needless to say, heavy traffic congestion aggravates this problem. Traffic congestion has various negative effects on travellers, businessmen, agencies and cities. According to TTI's 2011 urban mobility report, collected from 439 U.S. urban areas; Schrank et al.[1] discussed that one major factor is the amount of time and fuel wastage. According to this report, the top 15 urban areas comprise approximately 58% of the delay estimated for 2010, and the top 20 areas account for over 65% of annual delay. Furthermore, the urban districts with the populations of more than 3 million; accounts for 1.6 billion gallons (about 70% of the national estimation) of fuel wastage. As the congestion increases each year, the imposed cost will increase accordingly.

As shown in Table 1; the overall cost of congestion in urban areas is $100.9 billion in 2010 or an average of $713 per auto commuter. Therefore decreasing congestion level may lead to travel time and fuel consumption reduction and more importantly huge cost reduction. For decades, Intelligent Transportation Systems aim at improving the quality of urban transportation systems through employing various advanced technologies. Clearly, finding the efficient routes is an important objective in ITS which can be achieved through these two mechanisms: (1) Traffic information systems, (2) Vehicle routing protocols. Through the former mechanism real-time traffic information can be acquired, e.g. the density and speed of the vehicles; and the latter one makes use of the

Received date: Sep 11, 2013; Revised date: Oct 22, 2013; Accepted date: Dec 5, 2013
*Corresponding author. E-mail: s.yousefi@urmia.ac.ir
Copyright © 2013, China Association for Science and Technology. Electronic version published by Elsevier Limited. All rights reserved.
DOI: 10.1016/S1570-6672(13)60142-7

acquired traffic information in order to obtain the routes that the drivers should take.

Table 1 Congestion effect on the average commuter (for 439 U.S. urban areas)[1]

	Congestion statistic per auto commuter		
Population Group	Average Cost ($)	Average Delay (hours)	Average Fuel (gallons)
Very large areas	1083	52	25
Large areas	642	31	11
Medium areas	429	21	5
Small areas	363	18	4
Other urban areas	327	16	3
439 area average	713	34	14
439 area total	100.9 billion	4.8 billion	1.9 billion

Most of the current vehicle route suggestion systems use pre-set and offline city map along with the navigation data obtained from the GPS. Nadi and Delavar[2] enhanced the capabilities of these systems by utilizing offline and estimated traffic status which is mostly based on the previous historical data. However, these systems suffer from not employing the real-time traffic information in their routing process. As a result, their suggestions might not always be the optimal option due to the rapid change of traffic status. Recent progress in vehicular communications based on DSRC (Dedicated Short Range Communications) systems makes it possible to obtain traffic information in a real-time manner. With the information employed in vehicle route suggestion algorithms, more efficient routes will be attained.

The proposed approach includes two major phases. (1) Updated traffic information collection; (2) Vehicle route suggestion. In the first phase, we make use of the capabilities of inter-vehicle communications in order to design and implement the traffic Information system. In the second phase, through applying the collected traffic information, each RSU (Road-side unit) can find the best route towards the destination. For this purpose, we have proposed two algorithms called one-step vehicle path suggestion and step-by-step vehicle path suggestion algorithm.

The rest of this paper is organized as follows: Section 2 is aimed to give an overview of the related work. In section 3, the proposed mechanisms for both phases are explained. In section 4 the complexity analysis for both algorithms is precisely explained. In section 5 the simulation framework is described and the obtained results are evaluated. Eventually, section 6 sums up the study and summarizes our findings.

2 Related works

Since our work consists of two major phases, namely traffic information collection and optimal path suggestion process, some of the previous studies regarding both issues have been surveyed in the following.

2.1 Traffic information systems

In the field of ITS there are several ways to acquire the traffic information. Some of the most common methods include video and image processing, infrared sensors, magnetic sensors, inductive loop detector and piezoelectric Sensors. Mimbela and Klein[3] discussed the pros and cons of each these methods in details. Since those methods are not suitable enough in large scales and also cannot be applied in all weather and traffic conditions, some newer and more applicable methods have been proposed. Some of those novel methods are as follows.

In more advanced methods GPS (Global Positioning System) is used as a supplementary tool for traffic information collection. Dhingral and Gull[4] discussed the relationship between car speed, road capacity and road density. In this study a traffic model has also been proposed by observing and formulating historical traffic information and the developed model can be exercised to evaluate and estimate the number of vehicles in an urban environment. Skordyli and Trigoni[5] used stationary Aps (Access Points) in order to detect the vehicles' speed and the road density to help their protocol to plan navigation paths. However, the road density is hard to be detected using stationary access points. Therefore, the average vehicle speed detected by sensors has been considered and the number of the vehicles on a road is calculated according to the traffic flow theory Dhingral and Gull [4].

Vehicle to vehicle communications is also taken advantage in some recent methods. Kitani et al.[6] proposed a method which efficiently collects, retains and propagates traffic information via inter-vehicle communication with "message ferrying" method. In the proposed method, buses have been used as message ferries which travel along regular routes. For improving information propagation performance in low density districts, buses collect as much traffic information as possible from the vehicles in their proximity and disseminate this information periodically to the adjacent vehicles.

Khosroshahi et al.[7] proposed a new scheme for obtaining real-time traffic information in inter-urban routes which is mainly based on the combination of vehicular ad hoc networks with using IVC (Inter-vehicle communication) and VRC (Vehicle road-side device communication) and also conventional systems. An appropriate cost function and its parameters have also been defined. New parameters such as uncertainty are proposed for the first time to solve the route guidance problem using traffic information.

2.2 Vehicle path suggestion system

The classic SP (Shortest path) problem has been widely investigated in literature. Fan et al. [8] proved that when the utility function is linear or exponential, an efficient variant of Dijkstra algorithm can be employed in order to calculate the optimal route whenever the network is characterized as stationary and stochastic. Delling and Wagner[9] demonstrated that the standard SP algorithms are not capable of finding the minimum expected cost path when the network is non-stationary, dynamic, and stochastic and also the optimal route cannot be processed as a regular path. Instead, it has to be calculated based on a rule aka policy since there exist many dynamically changing parameters so that some policy-based decisions need to be made. Boutilier et. al.[10], Feldman and Valdez-Flores[11] investigated this issue. Bander and White[12] presented the AO* algorithm for a non-stationary, stochastic shortest path problem with terminal cost. For the non-stationary deterministic case, Hashemi et al.[13] investigated transit and parking time with time-varying cost for finding the shortest paths. Lim et al.[14] implemented an algorithm to determine the optimal path considering the delay probability distribution. Kim et al.[15], and Psaraftis and Tsitsiklis [16] elaborated the case when the congestion status of each edge is available for the driver. Ramalingam and Reps[17] applied this algorithm. Moreover, among the algorithms proposed for the DSP (Dynamic shortest Path) problem, the RR algorithm seems to be the most applied one. This algorithm which is a fully-dynamic DSP algorithm, updates the shortest paths incrementally. Buriol et. al.[18], Frigioni et. al.[19], and Fortz and Thorup[20] have employed this algorithm in their researches. It's been proved that the RR algorithm is an efficient approach for solving the DSP problem in terms of performance.

Goel et al.[21] investigated a time-constrained shortest path problem which tries to find a shortest path from a single source to all destinations within a time constraint. They have proved that this problem is NP-hard. In practice, due to huge number of vehicles, any path suggestion algorithm should be run in a distributed manner with tough timing deadlines. It means that each vehicle obtains the best possible path by itself without relying on any centralized entity. Besides, the speed of path suggestion algorithm is the main concern. Most of the above-reviewed works are essentially centralized and/or time-consuming. This indeed hinders their usage for practical usages in large urban environments. Many navigational products have been employed for several years and they attempt to discover the shortest paths for users with various sources and destinations[22, 23]. These systems often discover the shortest route according to the static geographic maps but the calculated route may become congested due to dynamic traffic alteration. Chen et. al.[24] employed a table of historical traffic information maintained using Intelligent Transport Systems for road density estimation.

The proposed path suggestion algorithm is different from the above-reviewed previous works in several aspects. First, it is on-demand (meaning that the route is found and suggested upon vehicles' request) despite many of the previous works which are proactive in nature for instance, the researches accomplished by Chen et al.[24] and Fan et al.[8]. Second, our approach does not depend on cell phone availability and the number of cell phones inside vehicles, despite cellular-based approaches such as that of Lin et al.[25] . Third, due to the lower cost of RSUs compared to cellular base stations, in our approach each RSU's interface (directional antenna) covers only one street while in cellular-based approaches the cellular base station may cover several streets simultaneously which causes inaccurate density estimation for each street.

The reason why we made use of the DSRC protocol and why this protocol is advantageous compared to other communication networks are in order. Both Wi-Fi and ZigBee standards work at unlicensed 2.4 GHz band, thus they easily get network interference or management overhead. In addition, their radio ranges are relatively shorter than DSRC. Other WPAN and WLAN techniques have obvious drawbacks. For example, Bluetooth (IEEE 802.15.1) has a 723 Kbps data rate and up to 10m radio range, but it only supports eight active devices and its network join time is too long (3s). Similarly, RFID (Radio Frequency Identification) techniques have problems in terms of radio range and low data rate. They can be used in vehicle identification and ETC (Electronic Toll Collection) application, but not for general IVC usages.

The latency of DSRC technology is about 200 micro seconds compared to cellular networks which is 1.5–3.5 seconds and satellite which is 10–60 seconds, its latency is considerably low. DSRC is especially designed for all ranges of ITS applications, thus normally it will have a better performance. Table 2 presents a comparison of the wireless sub-layer techniques that are available for current or near future IVC/ITS; which is partly taken from the research of Diao et al.[26].

Table 2 Comparison of considerable wireless sub-layer techniques [26]

Name	Max range	Max data rate	Licensed band	Mobility	Infrastructure necessity
DSRC	1 km	27 Mbps	Yes	Very high	Customizable
3G: UTRA-TDD	1 km	144 kbps (vehicle) 384 kbps (outdoor) 2 Mbps (indoor).	No	High	Yes
4G: WiMAX	50 km	40 Mbps (static) 14 Mbps (mobile)	Customizable	High	Yes
4G: LTE	100 km	100 Mbps (static and mobile)	Customizable	Very high	Yes

3 Proposed mechanism

The proposed approach comprises two major phases. (1) Traffic information collection,(2) Vehicle path suggestion. In the first phase, we employed capabilities of inter-vehicle communications in order to design a TIS in order to acquire updated traffic information. To obtain the traffic information including traffic density and vehicles' speed; directional antennas have been situated at the entrance of each street to collect traffic information which will be sent to the data centre. When a vehicle intends to set off, it sends a request to the nearest RSU in its vicinity. All RSUs are assumed to be connected to each other and to the data centre via a wired or a wireless network. In our simulation, we make use of WiMAX networks while other 2.5G (like GPRS and EDGE) or 3G cellular networks can be other feasible candidates. In the second phase, each RSU calculates the best route towards the destination. For this purpose, we have proposed two algorithms called one-step vehicle path suggestion and step-by-step vehicle path suggestion.

3.1 Phase 1 Traffic information collection approach

Through vehicular communications, it is possible to establish connections between vehicles and RSUs in an urban environment. In the proposed method each RSU obtains the traffic status of its corresponding road segments and this information will be transmitted to the data centre. The overall architecture is depicted in Fig. 1.

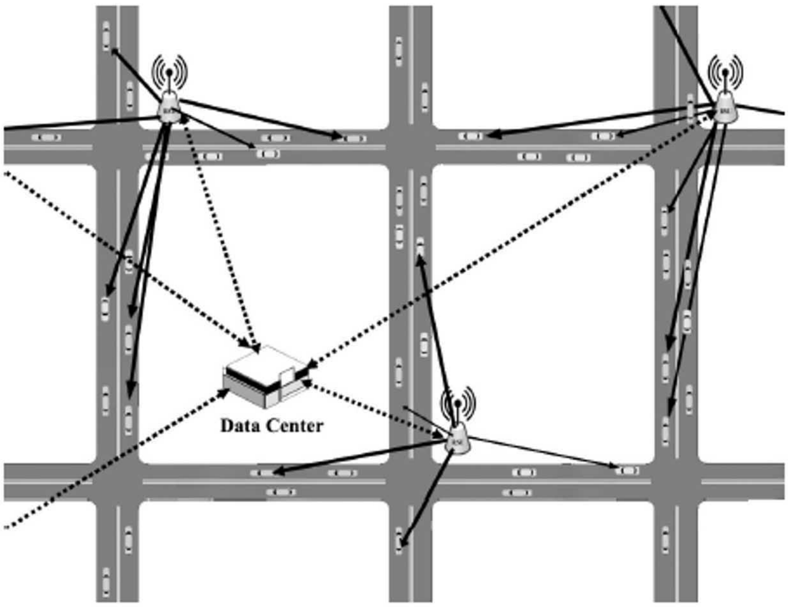

Fig.1 The communication between vehicles and RSUs (A), RSUs and the data center (B) Adapted from [27]

As shown in Fig. 1, RSUs are situated at specific locations of the city such as intersections. These RSUs are multi-interfaced which means that one RSU can be equipped with several wireless interfaces. IEEE 802.11 interfaces, cellular interface (GPRS), satellite interface and WiMAX interface are some of the typical examples of the utilized interfaces. Each RSU covers some specific streets. For example in a crossroad, four 802.11wireless interfaces and a WiMAX/GPRS interface will be required. Four 802.11 wireless interfaces are intended to be used for receiving the data sent by the vehicles and also for sending the traffic status from the data centre to the vehicles.

Furthermore, a WiMAX/GPRS interface is aimed to be used for transmitting data to or from the data centre.

3.1.1 Vehicles position detection using directional antennas

Directional antennas periodically send some information including street-ID, RSU-IP and RSU-port (i.e. socket address) of the respective street. Whenever a vehicle arrives at a street it receives this information from the directional antenna, then the vehicle will be able to specify its position and sends its location information along with the previous street-ID to the RSU. The information including the street's name/ID and RSU's IP address are used for establishing the further connections.

The reason that directional antennas have been applied is that, these antennas focus on one specific direction and are able to cover longer transmission and reception range and carry out more simultaneous transmissions and receptions within fewer steps. Ramanathan et al.[28] discussed the advantages of directional antennas in details. Another merit of using directional antennas is that since they only communicate with those vehicles that are moving along a certain street the signal interference would be decreased significantly.

3.1.2 Connection establishment

There are two types of connections in the proposed architecture: Firstly, the connection between vehicles and RSUs and secondly the connection between RSUs and the data centre. After receiving the current street-ID and the RSU's information, vehicles can establish a TCP or a UDP connection with the respective RSU, and transmit the ID of the previous street to it. The direction of the vehicles can be determined by using the previous street-ID and the current street-ID. Since RSUs are multi-interfaced and one interface is assigned to each street, that interface can simply recognize that from which street the data has been received. Then the RSU sends the ID of the previous and the current street to the data centre to update the density of each street. TCP is normally used to establish a connection between vehicles and RSUs; however, UDP is also a feasible option provided that some reliability measures are taken. The RSU socket address is transmitted by the directional antennas, which indicates that a RSU with a certain IP address is currently listening to the passing vehicles. The corresponding communication is labelled *A* in fig.1.

The communication between RSUs and the data centre has also been taken into account according to the proposed architecture. Although other alternatives may be used, in this study we used the centralized architecture. That is, All RSUs send the collected traffic information to a centralized data centre. It should be noted that if RSUs have adequate amount of storage and processing power they can play the role of data centre as well. Two most practical options for this step would be applying WiMAX (in the Point to Multi-Point mode) and cellular networks (such as GPRS and EDGE). In the simulations conducted in this study, we employed WiMAX connections between RSUs and the data centre; the corresponding communication is labelled *B* in fig. 1.

3.1.3 Data center functionality

Data centre has two main functions, (1) Updating the centralized traffic information database, (2) Informing all RSUs about the latest traffic status alterations. To accomplish the first function, after receiving the data through WiMAX/GPRS networks, the data centre updates the database. The data received from the RSUs include the followings:

① The ID/name of the previous street;

② The ID/name of the current street.

According to the previous and the current street ID, the data centre specifies the vehicles' direction and updates the database as follows: the data centre decreases the number of vehicles characterized with the previous street-ID by one and adds one unit to the number of vehicles associated with the current street.

To accomplish the second function, after updating the traffic information database, the updated data will be transmitted from the data centre to the RSUs through WiMAX/GPRS network. There are two possible methods in here:

① Instantaneous update: transmitting data to the RSUs immediately after being updated in the data centre;

② On-demand update: transmitting the updated data to the RSUs in case it is demanded by the vehicles.

In Instantaneous Update mode, as soon as the data is updated in the data centre it will be transmitted to the RSUs. The benefit of using this method is that the response time of the requests that the vehicles have sent in order to receive the updated data is considerably low. In the On-demand update, in case a traffic status request is made by the vehicles, the data will be transmitted from the data centre to the RSU and successively to the given vehicle. One of the outstanding advantage of the On-demand update is that notably small amount of data is exchanged between the RSUs and the data centre; however, in this mode, data is transmitted to the vehicles with slightly more delay compared to the Instantaneous update.

3.2 Phase 2 Vehicle path suggestion approach

For each requesting vehicle, we propose two algorithms aiming at suggesting a route towards its destination.

When a vehicle wants to get the optimal path towards the destination, it sends a request packet to the RSU and asks for the latest traffic information. Having received the vehicle's request, if the update process in RSUs has been set to on-demand mode, the RSU sends a traffic-information-update request to the data centre and the data centre sends back a reply packet containing traffic information update to the requested RSU. Afterwards the RSU sends the updated traffic information to the respective vehicle. If the update mode has been modified to Instantaneous, it conveys that RSUs already have the latest traffic information and no further traffic-update request would be necessary and all that RSUs have to do is sending the latest traffic update to the requested vehicle.

As soon as the vehicles receive the updated traffic information from the RSU, the process of finding the optimal possible route initiates. The routing process can be performed at RSU-side or the vehicle-side. The vehicle-side routing process is preferred, since it can help the network to reduce the congestion and the overhead of RSUs.

In order to find the optimal route, first we need to convert the city map into a graph, and then use a routing algorithm (such as a single-source and single-destination Dijkstra). Assume that the streets have the features shown in Fig. 2 below; i.e., they are two-way and letters A1, A2, B1, B2, ⋯, represent the streets names and status. For instance, A1 and A2 represent the traffic status between intersection 1 and intersection 2 respectively. A1 represents the traffic status from intersection 1 to intersection 2, and A2 represents the return path.

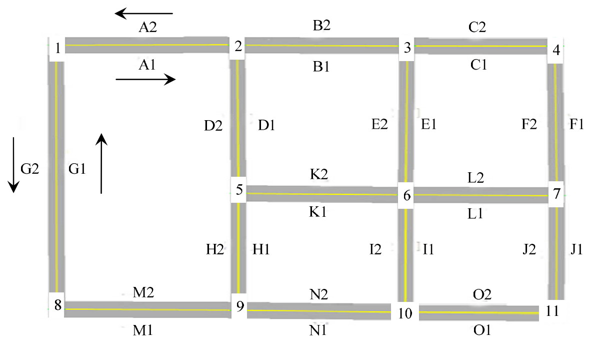

Fig. 2 An example of a street in an urban environment which can be converted into a graph

The number of vehicles present in each street will form the weight of the graph's edges (i.e. streets), and the intersections will be represented as the nodes. The details of weight function opted for this paper is mentioned in section 5.2. It should be noted that the process of finding the shortest route for the vehicles will be based on the latest traffic status of the streets and also the streets' length. Since the streets are two-way, the graph of the streets depicted in Fig. 2 will be a directed graph shown in Fig. 3 below.

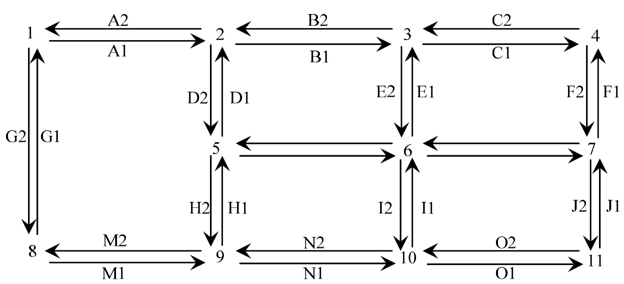

Fig. 3 The graph representation of the city map of Fig. 2

After determining the weights of the graph's edges, a standard shortest path routing algorithm such as Dijkstra or one of its variants can be applied to the network graph. We propose two approaches for the route suggestion problem as follows:

① One-step vehicle path suggestion;

② Step-by-step vehicle path suggestion.

3.2.1 One-Step vehicle path suggestion

Vehicles perform the routing process only once in one-step manner, by using the updated traffic information, and then moves towards its destination following the calculated route. Since the routing process has been done in one single step and the traffic status alters dynamically, the respective vehicle might get stuck in traffic congestion.

Assume that a vehicle is located in intersection 1 as depicted in Fig. 2, the route for this vehicle towards its destination will be determined based on the updated traffic status as follows:

1 → 2 → 3 → 6 → 10 → 11

Since the route suggestion process is operated in a non-dynamic way, it is probable that when the vehicle arrives at intersection 3 the street E2 gets congested so the vehicle will encounter heavy traffic jam and it'll get stuck there for a considerable amount of time. Having implemented the Dijkstra algorithm proposed by Wang and Ledley [29], this method finds the shortest route based on the weights of graph's edges.

3.2.2 Step-by-step vehicle path suggestion

In this approach similar to one-step method, vehicles use updated traffic information to navigate towards their destinations. However in contrast with one-step approach, at each intersection the routing process will be repeated, hence any traffic status alterations will be taken into account. Since the optimal path is updated at each intersection based on the newly updated traffic information, the probability of getting stuck in traffic jam significantly diminishes.

In step1 the path suggestion process initiates, and from step2 on; the path suggestion process restarts. In other words at the first intersection (step1) the process is identical to one-step method, and from the second intersection on, the process will be performed according to step-by-step path suggestion approach.

For example, assume that there is a vehicle at intersection 1 and its destination is intersection 11. The specified route for the first step of routing based on the data obtained from the data centre is as follows:

Step1 1 → 2 → 3 → 6 → 10 → 11

Since the routing process is being done for the first time at step1 (intersection1) (lines 1-5 in pseudo-codes of Fig. 4 and Fig.5), therefore the path suggestion process will be performed identical to one-step method. Since the pseudo-codes are illustrated by C language, the indices are initiated from 0.

Along this route, the vehicle continues to navigate towards intersection 2. As soon as it arrives at intersection 2; it requests the nearby RSU for the latest traffic information. Having received the information from the data centre, the vehicle implements a new routing process with the start point 2 and the destination 11. From intersection 2 the path suggestion process will be iterated (lines 5 on, in Fig. 4 and Fig. 5). The newly generated route at this step is as follows:

Step2 1 → 2 → 5 → 6 → 7 → 11

As soon as the vehicle arrives at intersection 5, this vehicle sends a request to the closest RSU to request for the latest traffic information. After receiving the updated traffic information from the RSU, the last routing process will be performed, this time with the current location 5 and the destination 11.

Step3 1 → 2 → 5 → 6 → 7 → 11

The path suggestion process will be iterated until the vehicle reaches its destination.

Step4 1 → 2 → 5 → 6 → 10 → 11

Step5 1 → 2 → 5 → 6 → 10 → 11

In step-by-step path suggestion method, the main challenge might be the loop creation issue. Assume that the start point is intersection 4 and the destination is 8, the routing process including step1, step2, and step *n* is as follows:

Step1 4 → 3 → 6 → 10 → 9 → 8

Step2 4 → 3 → 6 → 10 → 9 → 8

Step3 4 → 3 → 6 → 5 → 9 → 8

Step4 4 → 3 → 6 → 5 → 2 → 1→ 8

Step5 4 → 3 → 6 → 5 → 2 → 3 → 6 → 10 → 9 → 8

Step6 4 → 3 → 6 → 5 → 2 → 3 → 6 → 10 → 9 → 8

Step7 4 → 3 → 6 → 5 → 2 → 3 → 6 → 5 → 9 → 8

Step8 4 → 3 → 6 → 5 → 2 → 3 → 6 → 5 → 9 → 8

Step9 4 → 3 → 6 → 5 → 2 → 3 → 6 → 5 → 9 → 8

Step10 4 → 3 → 6 → 5 → 2 → 3 → 6 → 5 → 9 → 8

Considering the steps above a loop has been created in step 5. According to the normal behaviour of the shortest path algorithms, the decision in each step is made without having any knowledge of the previously travelled routes. We have proposed two approaches to sort out this problem namely, loop avoidance and loop detection.

In loop avoidance method, before the routing process is initiated, the weight of the streets through which the vehicles transit will be adjusted to infinity and the routing process will restart. For example in step 2, before the routing process restarts; the weight of street C in the graph will be set to infinity. In step 3, the weight of the streets C and E will be adjusted to infinity. Prior to routing process, this procedure is continuously performed until the vehicle reaches its destination. Because the adjustment of the weight values does not completely sort out the loop issue, it is essential to label those intersections (vertexes in the graphs) that the vehicles have already passed as previously-visited intersections. Then the vehicles won't pass the intersections again and the loop creation will be avoided this way. Through applying the previously-visited intersections as an effective parameter, loop creation issue can be resolved. Note that, the desired algorithm has to be modified according to some other criteria such as previously-visited intersections, besides the edges' status to achieve the above-mentioned purpose. The pseudo-code of step-by-step vehicle path suggestion improved by the loop avoidance mechanism is shown in Fig. 4.

```
    // Solution 1: Loop Avoidance mechanism

1       step:=0; // Path[step] equal current intersection
2       u := target;
3       Dijkstra(Graph, source);
4       Path[]<-S; // Sequence S copy to Array path
5       step:= step + 1;
6       while path[step] is defined:
7               for i:=0 to step-1
8                   Distance(path[i],path[i+1]):= infinity;// visited Edges (Streets)
9                   Distance(path[i+1],path[i]):= infinity;
10              end for;
11          Dijkstra(Graph, path[step]);
12          Path[from step to end]<-S; //add continue path from Sequence S to Array path, visited Vertexes (intersections)
    are in path [from 0 to step].
13          step:= step + 1;
14      end while;
```

Fig. 4 Step-by-step vehicle path suggestion improved by the loop avoidance mechanism

```
// Solution 2: Loop Detection

step:=0; // Path[step] equal current intersection
u := target;
Dijkstra(Graph, source);
Path[]<-S; // Sequence S copy to Array path
step:= step + 1;
while path[step] is defined:
        L:=1;
        while(L=1)
        Dijkstra(Graph,path[step]);
        If(loop detect)
            infinity one(or multiple) edge(s) in Distance as cause Loop
        else
                L:=0;
        end while;
    Path[from step to end]<-S; //add continue path from Sequence S to Array path, visited Vertexes
  (intersections) are in   path [from 0 to step].
    step:= step + 1;
end while;
```

Fig. 5 Step-by-step vehicle path suggestion improved by the loop detection mechanism

In the second method called loop detection, first the shortest route is generated and then the loop detection algorithm is applied. In case there is a loop, the value of a street or more than one street, is adjusted to infinity. The pseudo-code of step-by-step vehicle path suggestion modified using the loop detection mechanism is shown in Fig. 5.

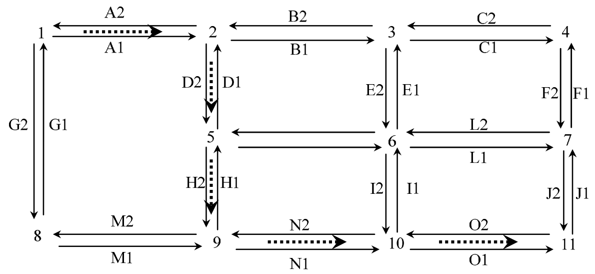

Fig. 6 Loop avoidance mechanism

As an example, in Fig. 6 a scenario of the step-by step vehicle route suggestion is proposed in which route avoidance mechanism has been employed. Assume that a vehicle wants to navigate from intersection 1 to intersection 11, it has to go through the following steps.

Step 0

Source=1, u=11, path = 1→2→3→4→7→11

Step 1 when the vehicle arrives at intersection 2

source = 2, u = 11, path = 1→2→5→6→7→11, weight streets A1 = A2 = ∞

Step 2 At intersection 5

Source = 5, u = 11, path = 1→2→5→9→10→11, weight streets A1 = A2 = ∞ and D1 = D2 = ∞

Step 3 At intersection 9

Source = 9, u = 11, path = 1→2→5→9→10→11, weight streets A1 = A2 = ∞ and D1 = D2 = ∞ and H1 = H2 = ∞

Step 4 At intersection 10

Source = 10, u = 11, path = 1→2→5→9→10→11, weight streets A1 = A2 = ∞ and D1 = D2 = ∞ and H1 = H2 = ∞ and N1 = N2 = ∞

And next step will be the last one therefore the vehicle navigates to intersection 11 which is the destination.

3.3 The system requirements and limitations

The requirements and limitations of our system are as follows. The equipment for the vehicular communications has to be situated at the appropriate locations, such as RSUs, OBUs and directional antennas. The RSUs have to be installed at the intersections, OBU on the vehicles and the directional antennas at the street's entrance. The system requires an acceptable penetration rate i.e. the number of vehicles that are equipped with the OBUs. If the penetration rate is not acceptable enough, this approach won't work efficiently. And last but not least, if the path suggestion process wants to be performed at the vehicle-side, the vehicles have to be equipped with the required processors and the user interface, in order to display the destination and routes.

4 Complexity analysis

The loop avoidance method and the loop detection method have been illustrated above. The number of vertices is depicted as n and the number of edges as e. In the complexity section we've assumed that the time complexity of Dijkstra algorithm is $O(n^2)$, although it can be improved to $O(e + n\log n)$ using an appropriate data structure[30].

4.1 One-step vehicle path suggestion

In this method the routing process will be performed only once at the first intersection, then the calculated route will be followed throughout the journey. Thus, in this mode the only time complexity is the complexity of the routing algorithm. In other words the time complexity of the loop detection will be $O(1)$. Since the routing process has been done in a single step hence no loop has been created. Considering the time complexity of Dijkstra algorithm mentioned-above, the overall complexity of one-step method will be $T(n) = O(1) + O(n^2) = O(n^2)$.

4.2 Step-by-step vehicle path suggestion

In this section we evaluate the time complexity of the step-by-step vehicle path suggestion mechanism under both loop avoidance and loop detection enhancements.

4.2.1 Solution1 : Loop avoidance

As it's been mentioned in loop avoidance section, the weights of the streets that the vehicles transit will be adjusted to infinity, and the previously-visited intersections are considered in generating the new route. The pseudo-code of loop avoidance mechanism has been illustrated in Fig. 4. The time complexity of this algorithm at each intersection in worst-case scenario equals to $O(n)$. Considering the loop avoidance pseudo-code, and the execution time of Dijkstra algorithm, the overall time complexity would be $T(n)= O(n)+O(n^2)+O(n) = O(n^2)$.

4.2.2 Solution 2: Loop detection

The pseudo-code of loop detection mechanism is illustrated in Fig. 5. In case there is a loop in the newly produced route in relation to the previously-visited routes, the weight values of all streets which led to loop creation have to be adjusted to infinity. Time complexity of loop detection method in the worst case will be $O(e^2)$; the time complexity of While loop in line 8 of Fig.5 will be $O(e)$. As a whole, the time complexity of loop detection method will be $O(e^3)$ (at each intersection).

As it's been illustrated in Fig. 7, the number of two-way streets will be approximately three times more than the number of intersections. In calculating the time complexity, $O(3n^3)$ is considered as $O(n^3)$; thus, the time complexity of the loop detection method will equal to $O(n^3)$.

Let's consider the execution time analysis:

(1) The time complexity of line 9 (i.e. Dijkstra algorithm) equals to $O(n^2)$.

(2) The time complexity of line 10 (loop detection step) equals to $O(e^2)$.

(3) The time complexity of line 11 (infinity edge(s)) equals to $O(e)$.

(4) The time complexity of line 8 equals to $O(e)*\left(O(n^2)+O(e^2)+O(e)\right)$.

(5) The time complexity of line 15 equals to $O(n)$.

Suppose that n=e, the overall running time would be $T(n) = O(n)*\left(O(n^2) + O(n^2) + O(n) \right) + O(n) = O(n^3)$.

The time complexity of the shortest route discovery algorithm such as Dijkstra, would be different depending on the implementation details. According to the analysis, it seems that in practice and for big cities it'd be better to manage the process in a hierarchical manner. The city should be divided into some limited geographical areas (segments) and intra-areas, hence the journeys can be managed internally using the segments' graph representation. Then inter-area journeys should be addressed using a graph whose vertices are the segments. This topic will be explored in our future work.

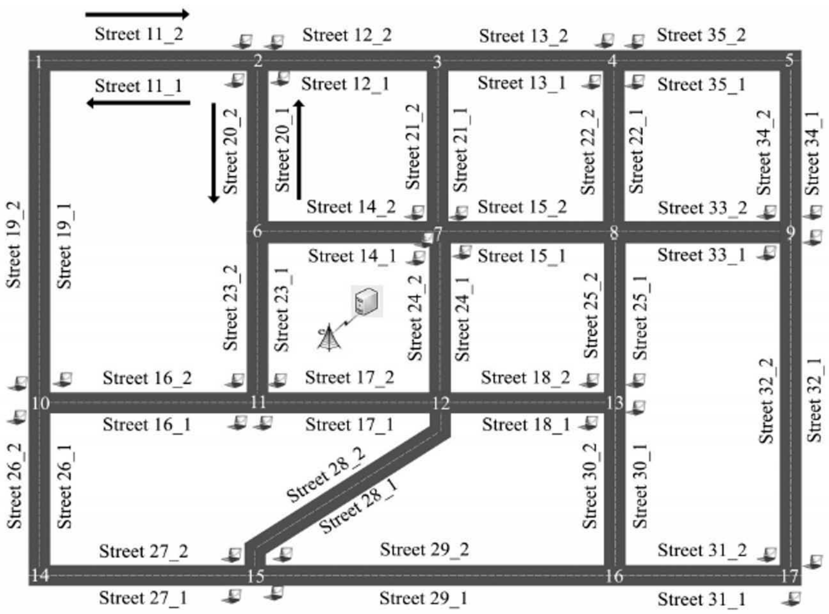

Fig. 7 The simulation environment

5 Simulation and evaluation of the results

The NCTUns 6.0 network simulator is utilized in this study in order to simulate the proposed approaches. As it's been depicted in Fig. 7, the simulation scenario consists of 25 two-way streets and 17 intersections. As it was described before, a real city is made up of several neighbouring scenarios such as the one depicted in Fig. 7; each scenario has a data centre which is allocated to that scenario.

5.1 Traffic information collection evaluation

This section is dedicated to evaluating the traffic information collection phase. The patterns of the data transmitted by the directional antennas can be seen in Fig. 8. The data is transmitted only to those vehicles located in street 27, so only vehicles 91, 105 and 106 will be at the receiving end.

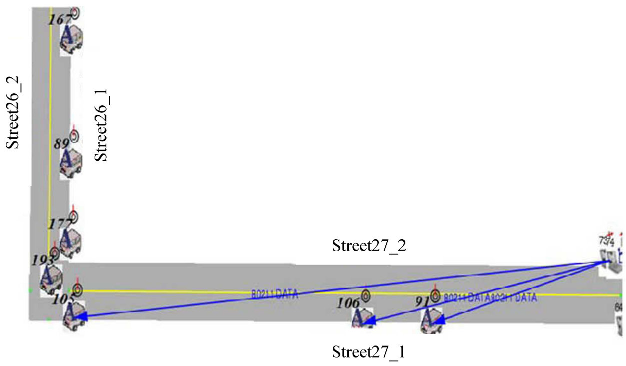

Fig. 8 The patterns of the data transmitted from directional antennas

First we have situated the RSUs at the intersections then 150 vehicles entered the traffic system. The simulation parameters are as follows:

- The simulation time: 380 seconds;
- The number of vehicles: 150;
- The average height of the buildings: 10 meters;
- The number of the streets: 25 (the names of the streets are ranged from 11 to 35).

The proposed traffic information collection approach aims at providing the vehicles with updated traffic information. Therefore one possible option to evaluate its functionality is to evaluate its precision in collecting the latest traffic information. For this purpose, at time intervals of 20, 40, 60,···,380, the data reported by the proposed method and the actual traffic status were evaluated and the error of the proposed method has been calculated accordingly. Equation 1 shows the error calculation formula.

$$AE_t = \frac{\left(\sum_{i=1}^{n} |PS_i - DR_i|\right) \times 100}{\text{number of vehicles}} \quad (1)$$

t = 20, 40, 60, End of Simulation (sec)

where, t stands for time (in seconds), n stands for the number of the streets, PS_i refers to the obtained value by the proposed method for the number of vehicles in street i at time t and DR_i stands for the number of the existing vehicles in the street i at time t. It should be noted that vehicle position detection is done using its coordinates. For instance, if the coordinate of a vehicle is within these intervals, 1 200< x <2 200 and 400< y <415, it means that this vehicle is in street number 15 and this vehicle should be counted in DR_{15}.

As shown in Fig. 9, this approach works with an acceptable error rate which remains below 15% all over the simulation execution. For instance, the error percentage of the proposed method at the 140th second is equal to 14%. A part of this error is due to collisions that occur while two or more vehicles transmit their messages simultaneously. Moreover the delay of TCP connection establishment process can also cause some error. It should be stressed that the amount of error can be tuned and customized by taking proper actions such as invoking collision avoidance techniques in the lower layer of protocol stack i.e., MAC layer or even in the application layer.

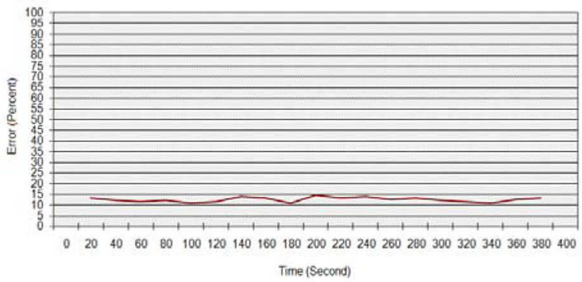

Fig. 9 The error rate of the traffic data collection system

5.2 Vehicle path suggestion evaluation

In vehicle path suggestion phase the optimal routes are calculated. In order to obtain the weight of each edge the following equation has been defined:

$$Weight_i = (w_1 \times SL_i) + (w_2 \times NV_i) + (w_3 \times NAV_i) \quad (2)$$

Where SL_i is the length of street i , NV_i is the number of vehicles in street i while the streets are not saturated and the traffic flow is fluent, and NAV_i is the number of extra vehicles in street i (i.e. beyond the street's capacity) and w_1, w_2, w_3 are the respective coefficients. The street capacity is the maximum number of vehicles which can simultaneously transit through the street at the allowed speed. This value is obtained considering the minimum allowable distance between moving vehicles according to transportation regulations. As an instance for a possible parameter setting, the value of w_1, w_2 and w_3 have been set to 0.45, 0.4, and 0.15 respectively. Since the street length and number of vehicles are two significant parameters in vehicle path suggestion process we allocated higher coefficients to them and for the other parameter, number of extra vehicles in each street, a lower coefficient has been assigned. In order to balance these parameters, 45% was assigned to the street length, 40% to vehicle number, and 15%

to the number of extra vehicles. Hence, the sum of these coefficients depicts the total street weight.

It should be noted that the above-mentioned parameter settings do not have any impact on the generality of the proposed approach and the parameters and weight function can be modified alternatively.

When the total weight of a street is calculated, the network graph will be generated. The graph used in the simulation environment has been demonstrated in Fig. 10. After generating the graph; both proposed path suggestion algorithms will be implemented.

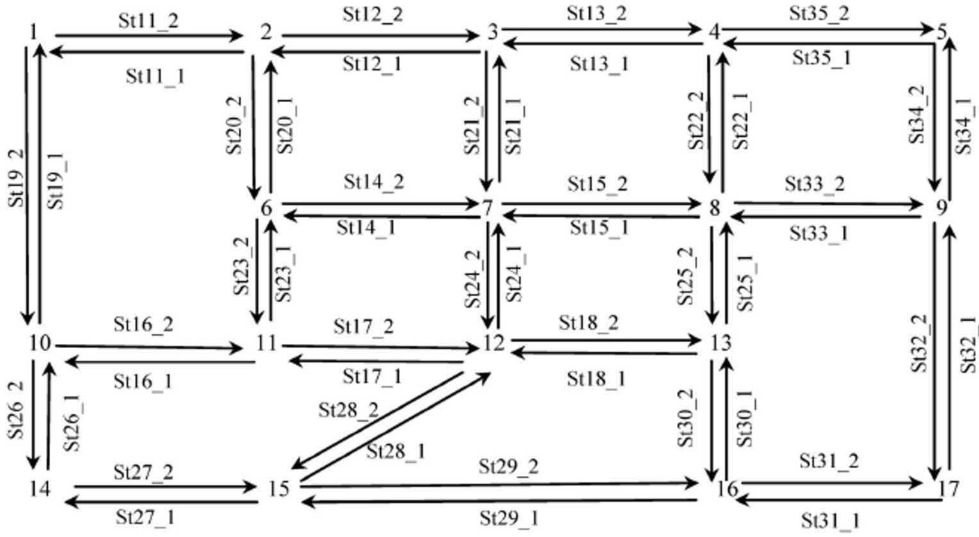

Fig. 10 The simulation graph

In order to evaluate the proposed methods, we randomly chose 24 vehicles out of 150 with different random start and destination points, and executed the simulation under both modes. Afterwards the performance has been evaluated in terms of travel time. The rest of parameters have been assigned, and they're identical to the simulation depicted in Fig. 9. The simulation results are shown in Fig.11. As this figure depicts; almost all vehicles are beneficiary if they opt to implement step-by-step route suggestion algorithm.

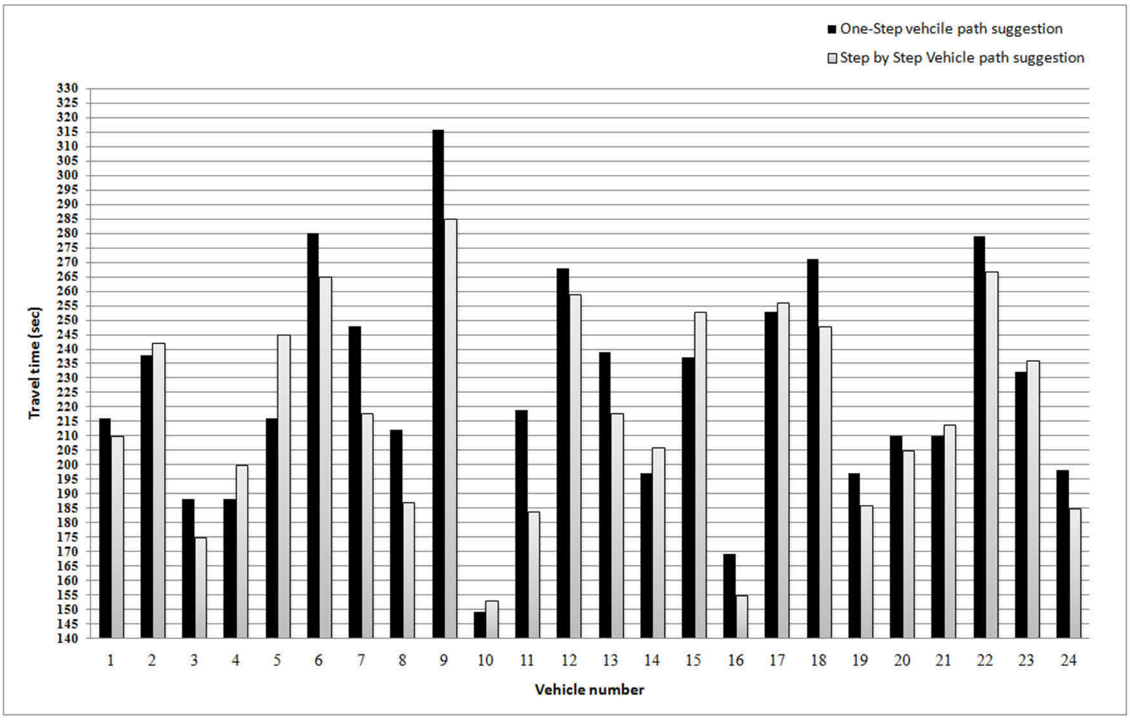

Fig. 11 The average travel time using one-step vs. step-by-step

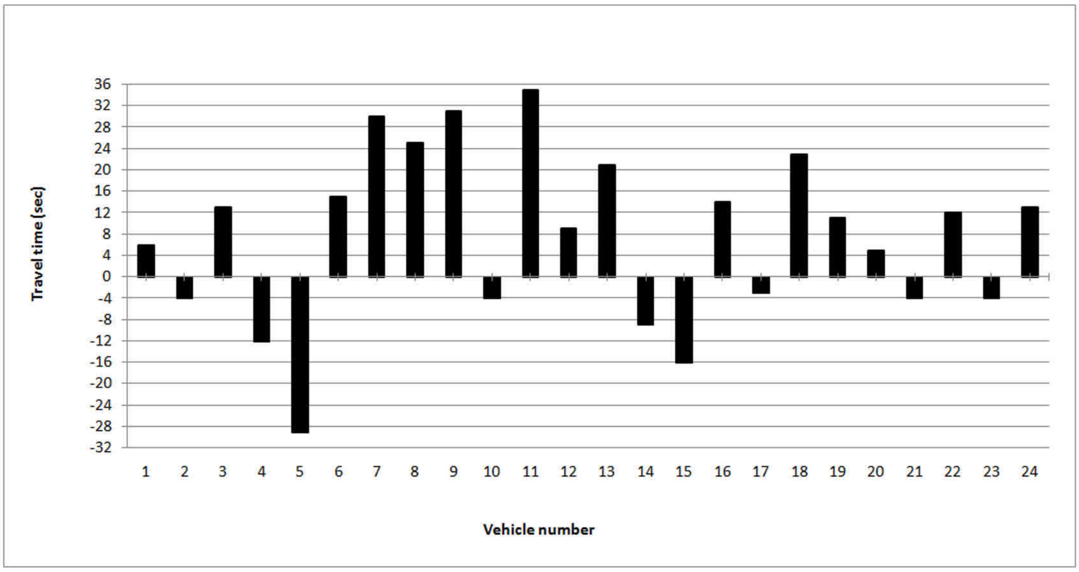

Fig. 12 The travel time reduction using step-by-step path suggestion vs. one-step path suggestion

Furthermore, the travel time reduction using step-by-step method compared to one-step method is demonstrated in Fig.12. By observing Fig.11 a probable question would be, why does the vehicle 20 and also vehicles 2, 10, 17, 21 and 23 have approximately the same journey time using either one-step or step-by-step routing methods? The rationale behind is that whenever the streets have almost the same density or the distance between the start and the destination point is short thus the travel time for both methods will be identical. Fig.11 indicates that in total 178 seconds have been saved in step-by-step mode compared to one-step mode with an average time reduction of 7.41 seconds for each vehicle.

Moreover, another question is that, why the amount of travel time reduction for vehicle 5 and also vehicles 4, 14, 15 is negative? In other words why the travel time using step-by-step method not only hasn't been decreased but also increased? The rationale behind this pertains to the parameters that were used for calculating the total weight of each street and also the fact that turning right, turning left, U-turn (180-degree) and direct movement have not been taken into account; these parameters are called trip quality parameters and have been investigated by Blue et al.[31]. The weight calculation enhancement considering other effective factors like trip quality in order to increase the precision will be the main focus of our future work.

In order to evaluate the performance of the proposed method under different traffic densities, we've implemented a simulation including 10 vehicles with different random start and destination points for both step-by-step and one-step modes. The simulation results show that, on average, the reduced amount of travel time using step-by-step mode compared to one-step mode will be increased along with density growth. The simulation results are illustrated in Fig.13. In this simulation, more notable effect for the Street Length parameter was taken into account in the routing process. In other words, this parameter had higher coefficient in the weight function (Eq. (1)) in order to alleviate the dominating effect of density in dense scenarios. Indeed, in Fig. 11 and Fig. 12 the coefficients are as follows: w_1= 45%, w_2= 40% and w_3= 15% while for Fig. 13, they are as follows: w_1= 50%, w_2= 35% and w_3= 15%.

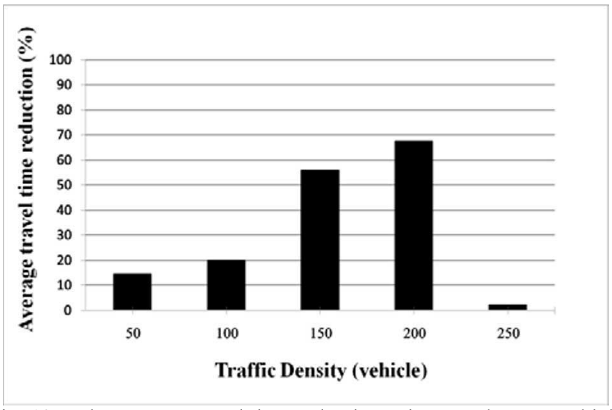

Fig. 13 The average travel time reduction using step-by-step vehicle path suggestion vs. one-step vehicle path suggestion for various traffic densities

Fig.13 depicts that when the streets include 50 or 100 vehicles, the travel time reduction in step-by-step method is not much different from one-step mode. However, if the density of the streets is increased to 150 or 200 vehicles, in this case, higher number of vehicles will reveal more notable difference between step-by-step and one-step modes. Noting Fig. 13 might raise the question that why the reduced amount of journey time is considerably low under the density of 250 vehicles. The rationale behind is that the streets are fully dense so that the routes generated via both modes of routing do not make a significant difference.

6 Conclusion

The route suggestion mechanism of this paper consists of two phases, namely updated traffic information collection and optimal path suggestion. In the first phase, RSUs inform the drivers about the latest traffic status in a considerably short amount of time. In the second phase, using the collected traffic information, vehicles are provided with the best possible route. Two algorithms have been proposed namely one-step path suggestion and step-by-step path suggestion algorithm.

Both traffic information collection and optimal path suggestion approaches have been simulated using NCTUns network simulator. After simulating this approach we evaluated the simulation results through error function. The results show the proposed traffic information approach can obtain recent density of the streets with an error rate below 15%.

In order to compare and evaluate the path suggestion approaches we've simulated both one-step and step-by-step methods. After comparing these approaches with static shortest path algorithms, we observed a significant improvement in travel time using either one-step or step-by-step route suggestion algorithms. As a whole, step-by-step approach outperforms one-step, in particular, we found that at very low and very high densities the improvement of step-by-step algorithm is not noticeable while in average density, the improvement is much more noticeable.

Since the time complexity of the proposed algorithms may increase dramatically for big cities, we intend to extend this work by performing the route suggestion in a hierarchical manner in future works. Furthermore, collision avoidance techniques will be developed in order to avoid message loss stemming from simultaneous transmissions.

References

[1] Schrank D, Lomax T, Eisele B. TTI's 2011 Urban mobility report powered by Inrix traffic data. Texas Transportation Institute Research Report, The Texas A&M University System, 2011.

[2] Nadi S, Delavar M R. Location-based service for In-vehicle route guidance with real time traffic information. The 12th World Conference on Transport Research (WSTR to WCTR) Proceedings, 2010:1-10.

[3] Mimbela L E Y, Klein L Y. A summary of vehicle detection and surveillance technologies used in intelligent transportation systems. Federal Highway Administrations (FHWA) Intelligent Transportation Systems Joint Program Office, Research Report, 2000 to 2007.

[4] Dhingral S L, Gull I. Traffic flow theory historical research perspectives. 75 Years of the Fundamental Diagram for Traffic Flow Theory: Greenshields Symposium Proceedings, 2008: 45-62.

[5] Skordyli A, Trigoni N. Delay-bounded routing in vehicular Ad-Hoc networks. In 9th ACM International Symposium on Mobile AD Hoc Networking and Computing (MobiHoc'08) Proceedings, 2008: 341-350.

[6] Kitani T, Shinkawa T, Shibata N, et al. Efficient VANET-based traffic information sharing using buses on regular routes. Vehicular Technology Conference (VTC Spring 2008) Proceedings, 2008: 3031-3036.

[7] Khosroshahi A H, Keshavarzi P, KoozehKanani Z D, et al. Acquiring real time traffic information using VANET and dynamic route guidance. 2nd International Conference on Computing, Control and Industrial Engineering (CCIE) Proceedings, 2011:3-13.

[8] Fan Y Y, Kalaba R E, Moore J E. Shortest paths in stochastic networks with correlated link costs. Computers & Mathematics with Applications, 2005, 49 (9-10): 1549-1564.

[9] Delling D, Wagner D. Time-dependent route planning, robust and online large-scale optimization. Lecture Note in Computer Science. 2009, 5868: 207-230.

[10] Boutilier C, Dearden R, Goldszmidt M. Stochastic dynamic programming with factored representations. Artificial Intelligence, Elsevier, 2000, 121 (1-2):49-107.

[11] Feldman R M, Valdez-Flores C. Applied probability and stochastic processes. Springer, 2010.

[12] Bander J L, White C C. A heuristic search approach for a non-stationary stochastic shortest path problem with terminal cost. Transportation Science, 2002, 36 (2): 218-230.

[13] Hashemi S M, Mokarami S, Nasrabadi E. Dynamic shortest path problems with time-varying costs. Optimization Letters, 2010, 4:147-156.

[14] Lim S, Balakrishnan H, Gifford D, et al. Stochastic motion planning and applications to traffic. Algorithmic Foundation of Robotics VIII, Springer Berlin Heidelberg, 2009, 57: 483-500.

[15] Kim S, Lewis M E, White C C. State space reduction for non-stationary stochastic shortest path problems with real-time traffic information. IEEE Transactions on Intelligent Transportation Systems, 2005, 6 (3): 273-284.

[16] Psaraftis H N, Tsitsiklis J N. Dynamic shortest paths in acyclic networks with Markovian arc costs. Operations Research, 1993, 41 (1):91-101.

[17] Ramalingam G, Reps T. An incremental algorithm for a generalization of the shortest-path problem. Journal of Algorithms, 1996, 21(2): 267-305.

[18] Buriol LS, Resende M G C, Ribeiro C C, et al. A hybrid genetic algorithm for the weight setting problem in OSPF/IS-IS routing. Networks, 2005, 46 (1): 36-56.

[19] Frigioni D, Marchetti-Spaccamela A, Nanni U. Semi-dynamic algorithms for maintaining single source shortest path trees. Algorithmics, 1998, 22:250-274.

[20] Fortz B, Thorup M. Increasing internet capacity using local search. Computational Optimization and Applications. 2004, 29:13-48.

[21] Goel A, Ramakrishnan K G, Kataria D, et al. Efficient computation of delay-sensitive routes from one source to all destinations. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2001) Proceedings, 2001:854-858.

[22] http://www.navman.com acc[OL]. 2014.

[23] http://www.papago.com.tw acc[OL]. 2014.

[24] Chen P Y, Guo Y M, Chen W T. Fuel-saving navigation system in vehicular Ad-hoc networks. IEEE 72nd Vehicular Technology Conference (VTC 2010-Fall) Proceedings, 2010: 1-5.

[25] Lin B Y, Chen C H, Lo C C. A traffic information estimation model using periodic location update events from cellular network. Communications in Computer and Information Science (CCIS), 2011, 135:72-77.

[26] Diao X, Mou K M, Li J J, et al. Inter-vehicle communication for the next generation of intelligent transport systems: Trends in geographic Ad Hoc routing techniques. Beylot A-L, Labiod H (Eds.), Vehicular Networks-Models and Algorithms, John Wiley & Sons, 2013,47-51.

[27] http://www.kurzweilai.net/driverless-vehicles-to-zip-at-full-speed-through-intersections acc[OL]. 2014.

[28] Ramanathan R, Redi J, Santivanez C, et al. Ad Hoc networking with directional antennas: A complete system solution. IEEE Journal on Selected Areas in Communications, 2005, 23(3): 496-506.

[29] Wang S P, Ledley R S. Computer architecture and security: Fundamentals of designing secure computer systems. John Wiley & Sons. 2013,254-255.

[30] Medhi D, Ramasamy K. Network routing: algorithms, protocols, and architectures. Morgan Kaufmann Publishers, 2010.

[31] Blue V J, Adler J L, List G F. Real-time multiple-objective path search for in-vehicle route guidance systems. Transportation Research Record, 1997, 1588: 159-167.