Adaptive traffic optimization

Ciprian Dobre, Andrei Ichimescu, Valentin Cristea
Department of Computer Science and Engineering
University POLITEHNICA
Bucharest, Romania
E-mails: ciprian.dobre@cs.pub.ro, andrei.ichimescu@gmail.com, valentin.cristea@cs.pub.ro

Abstract— **Wireless technologies can help solve traffic congestions in urban environments, where road infrastructures develop slower than the sometimes exponential growth in the number of cars in traffic. We present a system for traffic control and congestion avoidance. The system monitors traffic using cars acting as data collectors. It aggregates the monitoring data into a traffic model, further used by cars to dynamically adjust their routes. We propose novel metrics for traffic, and present evaluation results that show the capabilities of the proposed congestion avoidance model to correct traffic conditions.**

Keywords: congestion avoidance; mobile applications; traffic control.

I. INTRODUCTION

Traffic congestions are realities of urban environments. The road infrastructure capacities cannot cope with the rate of increase in the number of cars. This, coupled with traffic incidents, work zones, weather conditions [1], make traffic congestions one major concern for municipalities and research organizations.

The communication capabilities provided by wireless technologies and mobile devices offer opportunities for developing traffic control applications where cars and devices within the road infrastructure collaborate. Currently traffic lights organize the flow of traffic in intersections. Intelligent traffic lights, presented in [5], have the potential to adapt more easily to traffic conditions and/or provide priority to emergency cars, as well as safety to pedestrians and cyclists. The same principle can be easily applied for congestion avoidance. For example, a traffic light can have sensors (to collect data for traffic) and processing capabilities (to aggregate traffic data and take actions such as updating the time for the change of red or green lights or to log information for an economic pricing model to control congestions).

In this we extend the idea and propose a system that is able to solve congestions and control traffic using data collected from both sensors within the road infrastructure, and also from monitoring applications running inside cars. The immediate advantage of such an approach is that cars becomes data sources themselves, and can help collect traffic data from all places. The data monitored by cars is aggregated and developed into a model of the traffic. The model uses an innovative cost function, which includes several aspects designed to accurately describe traffic conditions. The model is further used to detect possible congestion problems, which are sent back to the drivers, and used to dynamically adjust their routes.

This approach allow the system to better control the traffic, such that to solve immediate congestions and realize a balance of the traffic on different roads. Drivers are motivated to participate in collecting traffic data, as they are the ultimate beneficiary of the end results. The route adaptation happens automatically, using a newly proposed path selection algorithm running within the mobile application inside the car.

Experimental results prove that the average time drivers stay in traffic is reduced, but also the amount of fuel they consume is also dramatically reduced (with an important positive ecological impact on the environment).

The rest of this paper is structured as follows. In Section 2 we present related work. Section 3 presents the model for congestion avoidance. In Section 4 we present details about the implementation of the model as an extension of a simulator designed for VANETs, experimental results. In Section 5 we give conclusions and present future work.

II. RELATED WORK

The problem of finding the optimum route for a destination in a crowded city has been previously approached in several research projects (e.g. [7], [8]). Similar to our approach, previous solutions also considered that cars are equipped with navigators able to receive feedback from traffic and dynamically adjust their routing paths. However, optimality is generally addressed by cars themselves, in relation to time (compute the routes such that to decrease the time cars spend in traffic) or distance (compute the routes such that to decrease the length of the paths considering that cars pollute). We argue that a more correct traffic decision should be made on a global scale. By letting cars control and choose their routes a situation where all cars see the same congestion roads and decide on the same alternative road – which will eventually congest the alternative road – is easily encountered.

Projects such as [7] approached the problem of computing optimal adjacent routes using learning techniques. After running on a route, the driver manually inputs the time spent in traffic, and in the future he/she will be able to select the best route depending on the average speed associated with known routes. However, such applications are limited, because they consider that traffic conditions are static. They also ignore events such as different weather conditions, accidents, etc, that are considered by many authors the roots of congestions [1]. Alternative solutions (e.g. [8]) process data received from various sensors within the road infrastructure (infrared cameras, asphalt integrated sensors, etc.) and compute the route dynamically, based on the traffic conditions (as opposed to using a classical Dijkstra algorithm). The authors present results showing that the alternative expert system is able to calculate a shortest path, but it also needs a large amount of data for computing the route. Also, the integration of sensors in every city intersection and/or road is costly, an aspect which can negatively influence the large-scale adoption of such a solution. At the moment, applications based on such types of sensors are used mainly on highways, and the services they provide are limited to traffic flow adjustment.

WAZE [4] is a free social traffic and navigation application that uses real-time road reports from drivers improve the navigation decisions. WAZE uses a social layer, where drivers work together to report and receive the most relevant traffic information available at any given moment. Our solution is similar in that we also rely on information collected by drivers. However, the computation and communications costs are reduced in our approach, as we are only interested in information related to traffic conditions. Unlike WAZE, we propose an approach where data is automatically collected by the mobile devices and aggregated, such that to take higher level decisions (such as the average speed cars are moving on a particular road segment). Also, by making use of alternative sources (sensors) and combining our approach with security solutions we previously proposed ([6]) we solve many of the problems that social networks have (intruders for example). We also propose several novel solutions. We use a number of distributed servers that are in charge of different traffic zones. Such an approach has advantages on communication (the traffic data is aggregated locally), and processing (data is collected from a smaller area).

III. A SYSTEM FOR TRAFFIC CONTROL

The system considers the following entities (see Figure 1): cars equipped with wireless devices and computation abilities, traffic lights acting as communication gateways between cars (wireless communication) and a centralized server (wired communication), and a server that constructs a traffic model and implements the algorithm for controlling traffic within the urban area.

We consider that *cars* are equipped with wireless devices where the driver runs a navigation application. This mobile application can send data about traffic (travelling speed, position) and receive back information used to dynamically adjust the travelling path. The monitored data is exchanged with the traffic light using close-range wireless communication. When the car is not able to directly communicate with a traffic light, it can either collect data about traffic conditions, and store it locally until it is able to wirelessly send it, or it could send the monitored data using 3G or WiMAX wireless communication technologies. Because of communication costs, we consider that most drivers will prefer the first collect-store-deliver approach.

Because they integrate communication and computing capabilities, the *traffic lights* are also called ‘intelligent traffic lights’ (the term proposed in [5]), or ITLs for short. Like cars, ITLs can also collect real-time data from traffic (for example, they could be equipped with sensors measuring the length of queuing cars). They aggregate data received from cars, and send back to the server all collected information (travelling speeds, number of cars queuing on the traffic light, etc.). To support the gateway role, ITLs have to be able to communicate with the server, as well as with cars in traffic. We consider this to be a realistic assumption, since most traffic lights today are already connected to a traffic control system. The system would only require the addition of limited-capability processing and wireless communication (and possible sensing capabilities as well, although it is not a strict requirement) equipments. The costs for such an investment could be supported by the gains in fuel consumption and street de-congestion. In addition, the inclusion of traffic lights in the system can support the adoption of different security models and mechanisms (we previously proposed such models based on fixed road entities in [6] and [9]).

The information is further aggregated by a centralized *server*, resulting in a model of the real-time traffic. In this model the graph of streets is augmented with data regarding the observed traffic. On the server-side this is further combined with past observations on the traffic. For the moment we consider a window of one hour of observations, and the server uses the data to compute the average congestion degree on each road segment for current hour. However, we plan to investigate this further and include a prediction model, where cars could also benefit from a predicted congestion value of roads ahead.

The obtained model of the traffic is used to implement a *congestion control mechanism*, where cars receive in support information about traffic conditions, and are able to dynamically adapt their travelling routes. Several aspects are considered. First, the system balances traffic by sending different data to different cars. If all cars would receive the same information, for example congestion on road A and good travelling conditions on road B, then all cars would deviate their routes to road B (thus, moving congestion from A to B). Secondly, cars are equipped with devices with limited computation and communication abilities. The driver expects results to be delivered as fast as possible to support its decisions (if the mobile application inside the car receives information about a congestion just after the driver already enters the congested road segments, the feedback will not be of much help anymore). To support this, cars receive information only for a limited area (as described in the next Section). To optimize communication, the server does not send information about all road segments. It computes all costs associated with road segments (the traffic model), but sends back only data associated with roads congested above a predefined threshold θ.

In summary, to support the traffic optimization, the proposed system implements three phases: traffic monitoring, data aggregation and the development of a traffic model, and traffic controlling using data feedback and dynamic route adaptation.

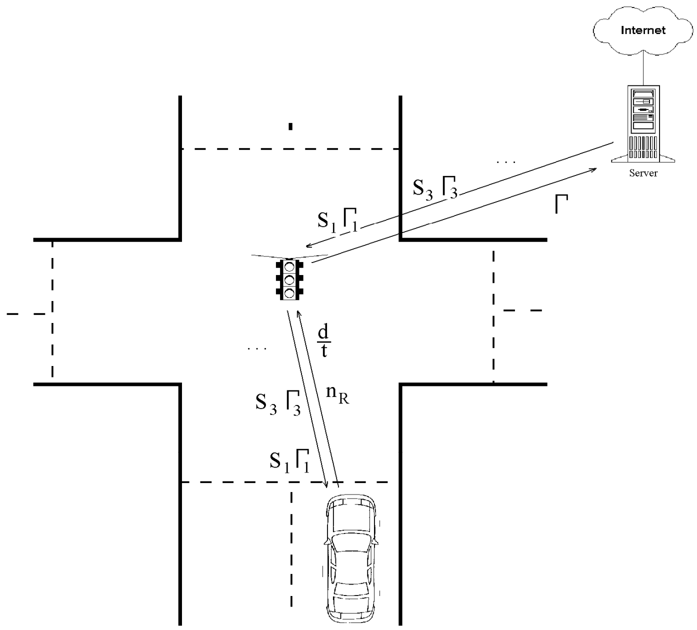

Figure 1. Entities within the system and their interactions.

3.1. Traffic model

The traffic model is a directed weighted graph, where nodes are intersections and the edges represent street segments. The weight associated with a street segment represents the cost of travelling on that particular road. To compute this cost we start from the observations of particular cars travelling on that road segment. These observations support the computation of a *congestion degree* (Γ), having a value between 0 and 255, depending on the load of the street segment: 0 represents an empty street, and 255 represent a heavily congested street. This weight is computed as follows:

$$\Gamma_{car} = \begin{cases} 255*(1-\frac{v_m}{v_{Max}}), & v_m \le v_{Max} \\ 0, & v_m > v_{Max} \end{cases} \quad (1)$$

where Γ_{car} represents the congestion degree observed by a particular car, v_m is the average travelling speed of that car on that road segment, and v_{Max} represents the maximum allowed travelling speed (depending on the local legislation) on that road segment.

The average speed (v_m) is computed considering the average travelling speed until the car reaches the ITL, coupled with the number of times the vehicles stops at the red light. This is correlated with the flow of vehicles passing through the intersection, measured in number of vehicles per second (it can happen that cars have problems passing through an intersection because of other connected congested roads). The formula to compute the average speed is:

$$v_m = v_{cruise} * (1 - \frac{n_R}{n_{RMax}} * \frac{F_{\min}}{F}) \quad (2)$$

where v_{cruise} represents the average cruising speed (between two consecutive 'stops' at the red light, or until the car crosses the intersection if the vehicle 'caught' the green light), n_R is the number of red light cycles the vehicle waited on, n_{RMax} is the maximum number of red light cycles a vehicle could wait on that particular road segment, F is the frequency of cars passing through the intersection, and $F_{\min}$ is the minimum frequency cars can pass by, as monitored for that intersection for the last hour. In this formula we

The average speed has a maximum value ($v_m = v_{cruise}$) when the traffic light stays green until the car crosses the intersection ($n_R = 0$). It has a minimum value ($v_m = 0$) when the car has to wait on many red cycles, and the frequency of cars passing through the intersection is minimum (both $n_R = n_{R\max}$ and $F = F_{\min}$ happen simultaneously). The average speed increases when the number of red cycles decreases, or when the frequency of cars passing through the intersection increases.

The main role of an ITL is to compute the congestion cost of the different monitored (or controlled by it) road segments. For every red-green light cycle an ITL collects the congestion degrees for all cars passing through the intersection. These values are used to compute an average value (after statistically eliminating possible outliers from observed data) for each road segment. It further sends this averaged value to the server (Γ denotes the average value of monitored Γ_{car} values). The server uses these values to compute the average congestion degree of each road segment for the current hour (denoted by Γ_m).

When the system is first installed, in the first week, the server will only collect the data. After the first week, the server will manage two separate traffic models: one represents the model of traffic as observed during the last week, and the other one represents the current traffic conditions. The differences (δ) between these two models show how much the traffic increased or decreased since last week. The relation between these terms is:

$$\Gamma = \Gamma_m + \delta \quad (3)$$

As previously described, to preserve bandwidth the server will send back to the cars Γ_m values for road segments only if δ passes a predefined θ threshold (or $|\delta| > \theta$).

3.2. Finding an optimal path

Using the data collected from traffic, the application inside the car runs an algorithm for finding the shortest path. It starts using only default (statistical if available) information (installed with the application) about road conditions. Each time it receives an update on traffic conditions it also dynamically adjusts the traveling path.

In our current implementation, the algorithm prioritizes the main roads to conduct the flow of traffic. The idea is that the application is much likely to lead to congestion (then solve it) if cars would be sent on smaller adjacent road segments (compared to when traffic is directed with higher probabilities on main roads, capable to sustaining it, inside a city). Also, as previously described, the traveling costs are computed based on observations mainly collected by cars. In this case the server will have a higher probability to compute a more accurate traffic model for main streets (where statistically more vehicles travel, so we expect to have more information) than for smaller streets (where in some cases we might even lack monitoring data altogether – traffic uncertainty). The lack of information can easily lead to situations where congestions observed on one road segment is solved by re-routing cars on other smaller road segments, where we assume there is no congestion (simply because we have no data about the traffic there). To solve such problems we propose prioritizing the traffic on higher road segments.

Initially the algorithm computes the shortest path using the traffic observations for last week. The computed path is the sum of three routes: (a) from the starting point to a main road, (b) from a main road until the destination, and (c) a path that connects the mains roads detected in (a) and (b).

Each time the car encounters an ITL, it receives updated observations for traffic and adjusts the travelling path accordingly. The update re-computes the corresponding path. If the car is on path (a), it receives updates for the smaller streets around it current position, and uses the data from last week for the rest of the roads to re-compute the shortest traveling path. If the car is on path (c), it receives only updates for the main roads (and uses the last known information about the other roads) to recomputed the path. Finally, if the car is on path (b), it receives updates only for smaller roads around its final destination to re-compute the path.

IV. Experimental results

The proposed system was evaluated using the VNSim simulator [2]. VNSim is designed as a realistic simulator for evaluating the performances of a wide-range of VANET technologies, ranging from wireless networking protocols and dissemination strategies to applications being developed over VANETs. The simulator is composed of two main models: a flow-based vehicular mobility model that considers driver personalities, traffic motions, realistic maps, and a wireless networking model, responsible with the simulation of the networking components and the communication protocols envisioned by a VANET system.

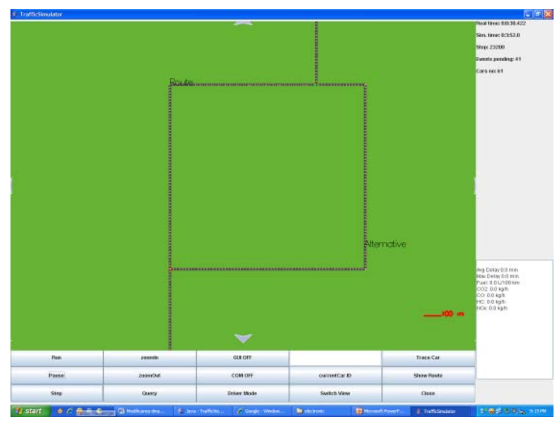

Figure 2. The first experimental scenario.

For evaluating the proposed security solution we extended VNSim with the capability to simulate cars travelling to different destinations, as well as intelligent traffic lights. As cars travel to their destinations, ITLs placed in various intersections broadcast periodically to cars the time until next change of light, how much time light is kept stable, and how many vehicles are queuing. This information is used by cars to compute the congestion degree (equation 1). The congestion degrees are propagates to the ITL, aggregated and the resulting average values are sent to a class emulating the behaviour of the centralized server. From there the previously described feedback mechanisms are used to dynamically adjust the travelling paths of cars.

For the evaluation experiments we considered two scenarios. The first one (in Figure 2) was used to evaluate the correct implementation of the proposed system within VNSim. In this case cars can only choose between two possible routes, so we were able to compare the results obtained in simulation with the expected results. These experiments were followed by a subsequent series of experiments (in Figure 3), designed to evaluate the performance obtained by the proposed system in terms of solving traffic congestions.

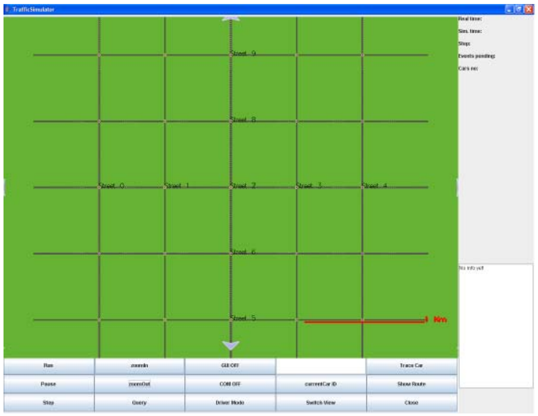

Figure 3. The second experimental scenario.

In the first experiment the two roads have each three different lanes per traveling direction. Cars travel from north to south, with an average flow of 400 vehicles / hour / lane. Initially all cars will chose the shortest path ("Route" in Fig. 2). As time passes cars will receive updates about the congestion will start preferring the second route ("Alternative" in Fig. 2). As more cars start choosing the alternative route, it will also become loaded. Eventually, this results to a balance of loads on both streets (as expected).

In the second scenario there are two main streets (each with three lanes per traveling direction). On each road cars travel across the map (there are cars traveling from north to south, south to north, east to west, and west to east). Every other road except for the main ones has two traveling lanes (they are used when the main roads become congested).

Using this scenario we conducted 18 different simulations. We executed 9 experiments using adaptive traffic lights, and 9 others using non-adaptive traffic lights. The non-adaptive ITLs have a constant cycle of succeeding red-green phases, while the adaptive-ITLs adapt the period red or green is on depending on the length of the waiting queue of cars. For example, when a street more circulated by cars intersects with another one less congested, then the adaptive ITL will hold the green color longer in case of the busy street.

Next, in each case we conducted three different series of experiments, varying the flow of cars: we used 350, 425, and then 500 cars / hour / lane. In each series of experiments we considered three situations: cars use the classical shortest path algorithm, cars run in the first week of the experiment (using only real-time data), and finally cars run in the second week of the experiment (this time cars can also week historical data collected over the past week).

Results for a simulation experiment include the number of cars reaching their destination, the average distance a car travels to its destination, average time a car travels to its destination, and the estimated fuel consumption. For the last result, the fuel consumption was computed using the model proposed by Akcelik&Besley [3].

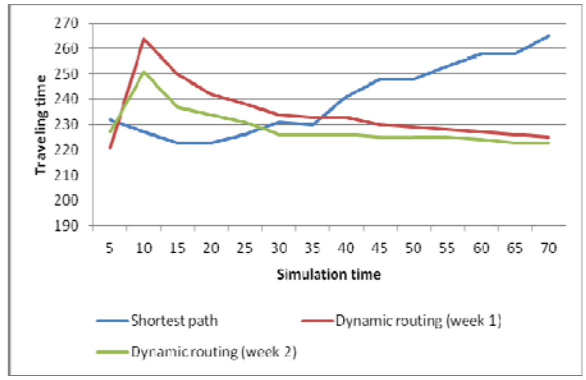

Figure 4. The average travelling time: adaptive traffic lights, 350 cars / hour / lane.

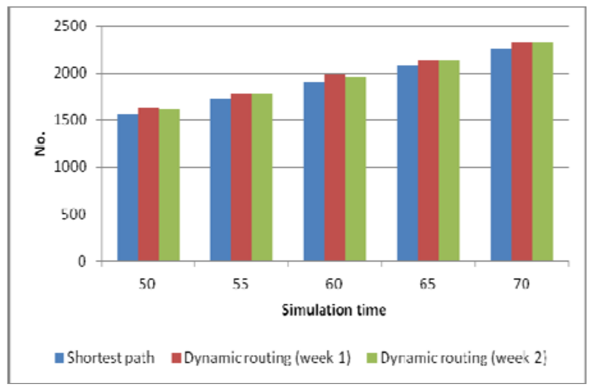

Figure 5. The number of cars reaching their destinations: adaptive traffic lights, 350 cars / hour / lane.

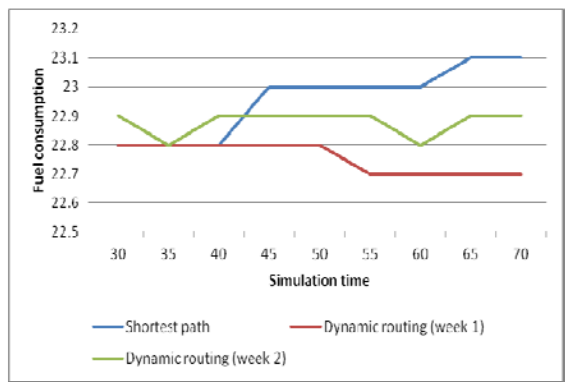

Figure 6. The average fuel consumption: adaptive traffic lights, 350 cars / hour / lane.

In the first series of experiment, using 350 cars / hour / lane we noticed an improvement on the number of cars reaching their destination (Figure 5) in the same amount of time from 2231 (in case of a classic approach) to 2324 cars (using the proposed system). The time necessary for cars to reach their destination (Figure 4) is also decreasing, as well as the average fuel consumption (Figure 6). The improvement obtained using dynamic routing reaches 22.9% in this case.

In the next series of experiments we gradually increased the number of travelling cars. In case of 425 cars / hour / lane, the proposed dynamic routing improves the rate of vehicles reaching their destination in the same amount of time by approximately 10%. Again, we noticed the same decrease in fuel consumption.

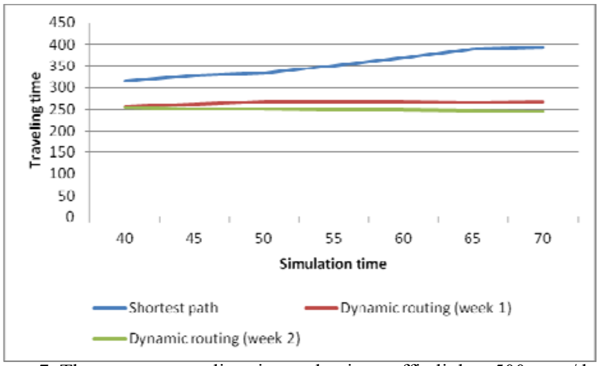

Figure 7. The average traveling time: adaptive traffic lights, 500 cars / hour / lane.

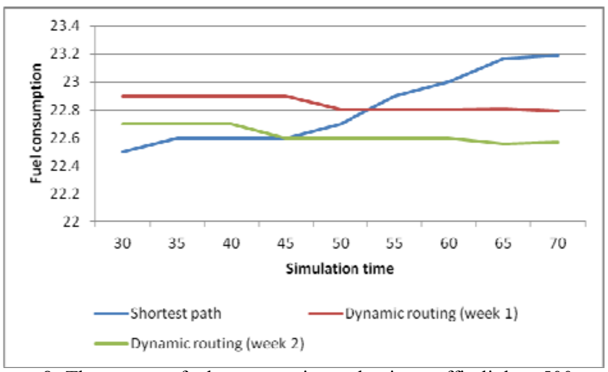

Figure 8. The average fuel consumption: adaptive traffic lights, 500 cars / hour / lane.

Finally, when using 500 cars / hour / lane, we obtained the best results in case of our proposed solution. The number of cars reaching their destination increased by approximately 18% compared to a classical shortest path routing, and the fuel consumption was decreased from an average 0.765 l to 0.746 l per travel. In average each car spent with 37.6% less time in traffic compared to travelling using the classic navigator.

V. Conclusions

We presented a solution for traffic control and congestion avoidance in urban environments. Previous studies showed the important role played by the transport infrastructure in modern economies. It is estimated that only in Europe traffic congestion affect approximately 10% of the existing transport network, with significant financial implications. Such reports recognize that the development of new infrastructures is not a reliable solution as compared to investing in technologies based on intelligent transport systems (ITS) and Vehicular Ad-Hoc Networks (VANETs).

In this we proposed a solution for constructing a traffic model to solve congestions. Traffic data is collected from the road, aggregated and provided as feedback to cars. The solution uses cars to collect traffic data and several ITLs that are able to aggregate and take decisions as how to influence the routes the cars are driving. Whenever a road segment starts to provide lower average speeds for vehicles passing through, a routing algorithm provides alternatives routes, less congested and providing better times to reach destinations.

We evaluated the model in a series of simulation experiments. We demonstrated that the average time needed for vehicles to reach their destinations registered a significant decrease of up to almost 40% compared with the time needed for vehicles to reach their destination using predefined static routes. The average fuel consumption also registers economies up to 1 liter per 100 kilometers, which is a major advantage considering the fact that in a city with millions of vehicles the fuel saving can be quite high. The total emissions also show a high decrease because of less fuel consumption and the fewer accelerations and brakes that vehicles need to apply. The emissions' decrease varies from 14% to 40%.

VNSim's official site and extended details on these experiments are available at http://cipsm.hpc.pub.ro/vanet.

Acknowledgment

The research presented in this paper is supported by national project: "TRANSYS – Models and Techniques for Traffic Optimizing in Urban Environments", Contract No. 4/28.07.2010, Project CNCSIS-PN-II-RU-PD ID: 238. The work has been co-funded by the Sectoral Operational Programme Human Resources Development 2007-2013 of the Romanian Ministry of Labour, Family and Social Protection through the Financial Agreement POSDRU/89/1.5/S/62557.

References

[1] US Department of Transportation, "Traffic Congestion and Reliability: Trends and Advanced Strategies for Congestion Mitigation", Final Report, September 1, 2005, accessed February 2nd, 2012, from http://ops.fhwa.dot.gov/congestion_report.

[2] A. Gainaru, C. Dobre, V. Cristea, A Realistic Mobility Model Based on Social Networks for the Simulation of VANETs. In: *Proc. of the IEEE 69th Vehicular Technical Conf.* (VTC Spring 2009). Barcelona, Spain, pp. 1-5, 2009.

[3] R. Akcelik, and M. Besley, Operating cost, fuel consumption, and emission models in aasidra and aamotion, In: *Proc. of 25th Conf. of Australian Institutes of Transport Research* (CAITR 2003), University of South Australia, Adelaide, Australia, December 2003.

[4] I. Lequerica, M. García Longaron, P.M. Ruiz, Drive and share: efficient provisioning of social networks in vehicular scenarios, *IEEE Communications Magazine*, 48(11), 2010, pp. 90-97.

[5] V. Gradinescu, C.Gorgorin, R. Diaconescu, V. Cristea, L. Iftode, Adaptive Traffic Lighs using Car-to-Car Communication, In: Proc. of VTC2007-Spring, Dublin, Ireland, pp. 21-25, 2007.

[6] S. Mazilu, M. Teler, C. Dobre, Securing Vehicular Networks Based on Data-Trust Computation. In: Proc. of *3PGCIC 2011*, Barcelona Spain, pp. 51-58, 2011.

[7] B. Liu, Using Knowledge to Isolate Search in Route Finding, In: *Proc. of 14th International Joint Conference on Artificial Intelligence (IJCAI-95)*, Montréal, Québec, Canada, 1995, pp. 119-124.

[8] G. Eggenkamp, L.J.M. Rothkrantz, Intelligent dynamic route planning, In: *Proc. of the 13th Belgium-Netherlands Conference on Artificial Intelligence*, Amsterdam, 2001, pp. 381-388.

[9] C. Gosman, C. Dobre, V. Cristea, A Security Protocol for Vehicular Distributed Systems. In: Proc. of *SYNASC 2010*, Timisoara, Romania, pp. 321-327, 2010.