2016 IEEE Symposium on Computers and Communication (ISCC)

Real-Time Path Planning to Prevent Traffic Jam Through an Intelligent Transportation System

Allan M. de Souza[1], Roberto S. Yokoyama[1,3], Guilherme Maia[2], Antonio Loureiro[2] and Leandro Villas[1]
[1]Institute of Computing, University of Campinas, Brazil
[2]Computer Science Department, Federal University of Minas Gerais, Brazil
[3]Federal University of Technology - Paraná, UTFPR, Cornélio Procópio, Brazil
Email: {allanms, sadao}@lrc.ic.unicamp.br, {jgmm, loureiro}@dcc.ufmg.br, leandro@ic.unicamp.br

***Abstract*—Congestion is a major problem in large cities. One of the main causes of congestion is the sudden increase of vehicle traffic during peak hours. Current solutions are based on perceiving road traffic conditions and re-routing vehicles to avoid the congested area. However, they do not consider the impact of these changes on near future traffic patterns. Hence, these approaches are unable to provide a long-term solution to the congestion problem, since when suggesting alternative routes they create new bottlenecks at roads closer to the congested one, thus just transferring the problem from one point to another. With this issue in mind, we propose an intelligent traffic system called CHIMERA, which improves the overall spatial utilization of a road network and also reduces the average vehicle travel costs by avoiding vehicles from getting stuck in traffic. Simulation results show that our proposal is more efficient in forecasting congestion and is able to re-route vehicles appropriately, performing a proper load balance of vehicular traffic.**

I. Introduction

Traffic jam is a problem in most large cities worldwide. Usually, it is caused by the sudden increase in the number of vehicles on roads during peak hours and bottlenecks in the transportation infrastructure. Even with the continuous improvement of the urban transportation system, the number of vehicles tend to increase as the economy develops. Therefore, road traffic congestion becomes a recurrent problem. Several negative impacts for society are associated with congestion, for instance, economic losses, productivity reduction, and the increase of carbon dioxide emissions as drivers get stuck in traffic [2], [9].

In the past few years, advancements in wireless communication technologies and the development of vehicular network standards paved the way for the implementation of Intelligent Transport Systems (ITS) [7], [17]. ITS is a comprehensive solution for real-time traffic management that relies on data collection from vehicles, road sides units (RSU) and other sensors, which are entities that can interact and cooperate among themselves, creating a vehicular network [13], [15], [16]. An ITS application must detect, control and reduce congestion based on on-line data that describes traffic patterns, such as, density, speed, travel time, geographic position of vehicles and current time. To accomplish this goal, however, the main challenge is how to forecast congestion and re-route vehicles appropriately by considering the short-time impact on future traffic in an area of interest (AoI) [2], [5], [8], [12], [17].

The demand for a solution to avoid traffic congestion is very clear. The most popular application toward this end is Waze[1], which has gained increasing attention in recent years. However, Waze is far from achieving the ITS goals, since it depends exclusively on information posted by people using their smartphones. Moreover, routes suggested to users are based exclusively on shortest-path algorithms, without taking into account the impact re-routing has in future traffic conditions. Other solutions, which are based on vehicular networks, can be found in the literature [2], [5], [12], [17]. These solutions focus on solving only part of the problem, such as, Bauza et al. [2], which focus on congestion detection while Pan et al. [12] focus on re-routing. Moreover, there is a solution specific to highway scenarios [14].

[1]https://www.waze.com

In this paper, we go further and propose an intelligent traffic application called CHIMERA (**C**ongestion avoidance throug**H** a traff**I**c classification **ME**chanism and a **R**e-routing **A**lgorithm) based on vehicular networks. CHIMERA is able to detect when a congestion is forming and balance the traffic in a clever way to distribute the density of vehicles in order to avoid more congestion in the near future. Therefore, CHIMERA is aware of the traffic conditions of an AoI using information collected periodically from vehicles. Using this data, CHIMERA regularly ranks each street to a congestion level. Applying a re-routing algorithm, which considers the congestion levels, CHIMERA maintains a smooth traffic flow in the AoI.

To demonstrate the efficiency of our proposal, we compare CHIMERA to some of the main solutions found in the literature by means of simulations. When compared to three well-known solutions – WithRouting [5], DSP and RkSP [12] – we show that our proposal is more efficient in forecasting congestion and re-routing vehicles appropriately, performing a proper load balance of vehicle traffic.

This work is organized as follows. Section II discusses the literature related to congestion minimization in urban environments. Section III describes CHIMERA. Section IV presents the results of the performance evaluation. Finally, Section V concludes the study and discusses some future work.

II. Related Work

Traffic jam, caused by unbalanced traffic flow or deficiencies in the transportation infrastructure, can lead to late arrivals and additional cost for drivers and becomes a major problem in the urban mobility. However, the cost due to traffic congestion can be reduced by path planning with congestion avoidance.

Bauza et al. [2] propose CoTEC (COoperative Traffic congestion detECtion), a novel cooperative vehicular system based on Vehicle-to-Vehicle (V2V) communication to detect traffic congestion using fuzzy logic. CoTEC uses messages in order to inform all vehicles about traffic conditions and detects a potential congestion condition locally at each vehicle.

Therefore, upon detecting a traffic jam, each vehicle broadcasts its own estimation about the traffic jam and, then, with all estimations, vehicles collaboratively detect and characterize the road traffic congestion. However, the mechanism employed to identify the traffic condition can cause an overload in the network due to the periodic beacon messages and the local estimations disseminated by all vehicles. Furthermore, despite being able to detect congestion, no mechanism to minimize or control the traffic jam is presented. In addition, CoTEC was proposed to operate exclusively on highway scenarios.

Pan et al. [12] propose a centralized system to acquire, in real-time, the vehicle geographic position, speed and direction as a means to detect traffic jams. Once detected, vehicles are re-routed based on two different algorithms. First, *Dynamic Shortest Path* (DSP), which routes vehicles using the shortest paths that also have the lowest traveled time. However, one shortcoming of this algorithm is the possibility to move the congestion to another spot. Second, *Random k Shortest Paths* (RkSP), which randomly chooses a route among k shortest path routes. The goal of this algorithm is to avoid switching congestion from one spot to another by balancing the re-routed traffic among several paths. This scheme does not implement a real-time mechanism to detect congestion as it occurs, only detecting it during the next re-routing phase.

Brennand et al. [5] propose an ITS based on a set of RSUs distributed in order to provide full coverage of a city. Each RSU is responsible for managing vehicles and detecting congestion only within its coverage area. Moreover, the proposed ITS includes a congestion control mechanism, which periodically performs the re-routing of all vehicles, so they do not go through congested areas. Similarly to Pan et al. [12], this scheme does not detect congestion as soon as its occurs, since it only detects traffic jams during the next re-routing phase.

The related proposals present some limitations such as, high communication overload, limited routing distance and no real-time mechanisms for congestion detection. Therefore, in this work, we go a step further and propose a congestion avoidance system through a traffic classification and a re-routing algorithm, which addresses the limitations presented by existing solutions.

III. Proposed Solution

In this section, we describe CHIMERA, an ITS composed of three main procedures, which we describe in detail in the following. Section III-A presents the model used to represent the urban streets and how the data collection procedure from vehicles works. Section III-B describes the technique employed to classify the congestion levels at every street. Finally, Section III-C presents the re-routing algorithm that balances the traffic density according to the congestion levels of each street.

A. Road network and communication model

A directed and weighted graph $G = (V, E)$ represents the road network, where the set V corresponds to the set of intersections (vertices), while the set $E = \{e_1, e_2, ..., e_i\}$ corresponds to the set of road segments (edges). Moreover, $W = \{w_1, w_2, \ldots, w_i\}$ is a set of weights representing the weight of each edge in E. Let $\nu = \{v_1, v_2, \ldots, v_j\}$ be a set of vehicles, $R = \{r_1, r_2, \ldots, r_j\}$ a set of routes for each vehicle in ν, and C the set of congested roads, where $C \subset E$. Every vehicle v_j periodically sends a message $mgs^j \leftarrow \{position^j, speed^j, route^j, destination^j\}$ to the nearest RSU. At every time interval t, the RSU computes the average speed, s_i, and the total number of vehicles n_i at each edge e_i. Then, the RSU updates the edges' weights $w_i \leftarrow (s_i, n_i, ms_i)$, where ms_i is the speed limit for e_i. Based on these weights, the RSU classifies every road in E and creates the set of congested roads C. For each edge $e_i \in C$, we define an AoI_i to prioritize every vehicle v_j close to e_i. For each vehicle v_j with a route r_j, a new route is calculated to maintain a smooth traffic flow in the AoI.

CHIMERA assumes that a RSU covers the scenario. Moreover, each vehicle v_j periodically sends a message mgs^j to the RSU by means of a single hop long-range communication process, such as LTE or 4G. Furthermore, at each time interval t, the RSU computes the total number of vehicles n and the average speed s for each road, and updates the edges' weights W according to Equation 1. These procedures are shown in Algorithm 1 (Lines 1–15).

The weight equation (see Equation 1) was modeled to be inversely proportional to the traffic condition in the road and to distribute the weights in the interval of $[0, 1]$ $w_i \in [0, 1]$. This way, the better the traffic condition on a road is, the lower its weight value will be. Thus, congested roads have higher weights than free-flow roads. Equation 1 is defined as follows:

$$w_i = 1 - \frac{s_i}{n_i \cdot ms_i} \mid n_i > 0 \tag{1}$$

where s_i, n_i and ms_i are defined according to road e_i, and denote the average speed of vehicles, total number of vehicles and the maximum speed of vehicles on e_i, respectively.

B. Data processing and congestion detection

Due to the complexity of handling the huge amount of information, there are several approaches to detect traffic congestion [2], [5]. Many studies that focus on road status identification use values such as the total and average speed of vehicles to detect traffic congestion. In these cases, the most applied technique to classify the level of traffic congestion is fuzzy logic [1], [2]. CHIMERA differs from these studies because it employs the k-Nearest Neighbor algorithm (KNN) [11], which is a more suitable technique to classify congestion levels on roads, besides being more simple.

In general, the k-NN algorithm uses a sample dataset to train the classifier. For that purpose, we built a synthetic dataset based on the Highway Capacity Manual (HCM) [4]. The HCM contains concepts, guidelines and procedures to measure the capacity, performance and quality of traffic, based on the speed and density of vehicles on the roads. The dataset was built based on the Level-Of-Services (LOS) present in HCM. The LOS represents a quality measurement used to describe the operational conditions within a traffic stream. With the training dataset we were able to identify four traffic conditions, where each one of them is based on the speed limits presented in the LOS of HCM. Furthermore, to decrease false positives, we relied on the road density in combination with the average road speed to define a traffic condition. The density is based on the percentage of vehicles in the road according to its maximum capacity. Thus, according to our dataset, the congestion classification is defined as *free-flow* = 0, *slight congestion* = $1/3$, *moderate congestion* = $2/3$ and *severe congestion* = 1. The k-NN algorithm implemented in CHIMERA uses the average road speed and the density of vehicles in the road as input parameters. The output of the algorithm is the traffic condition defined by our training dataset. Figure 1 shows an example, where CHIMERA just coletected the information of all vehicles and already classified the roads. Futhermore, a congested road is detected and a AoI is defined. Moreover, a vehicle within the AoI will pass by a congested road, therefore a new route is calculated to the vehicle to avoid to pass through a congestion.

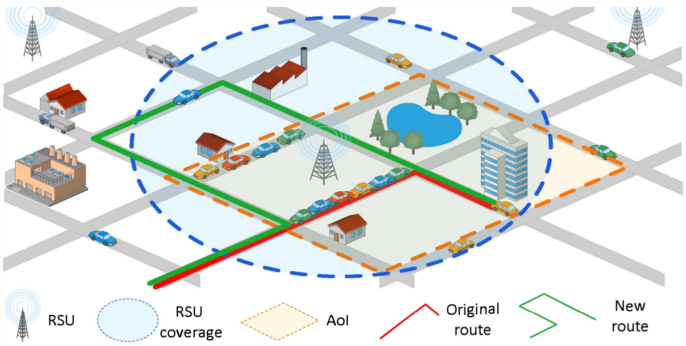


Figure 1. The main CHIMERA entities: RSU, AoI, congestion, vehicle and routes. A vehicle inside the AoI that will pass through the congestion is re-routed by the system.

The steps to perform traffic congestion detection and classification are shown in Algorithm 1. CHIMERA makes the classification of the roads which are under each RSU coverage (see Algorithm 1 – Lines 9 and 10). Thereafter, each RSU knows the route of each vehicle in its own coverage, so each RSU can determine which vehicle will go through a congested road (Algorithm 1 – Line 12 to 14). On the other hand, for each congested road $e_i \in C$, an AoI is defined to prioritize vehicles v_j that have a higher probability of passing through this congested road in a near future. Thereafter, when the RSU detects that a vehicle is inside an AoI of any congested road $v_j \in AoI_i$ and this vehicle will pass through that congested road $r_j \subset C$ (Algorithm 1 – Line 15), an alternative route r_{new}needs to be calculated to avoid the congested one. In Figure 1, we show that the moment when the RSU detects a vehicle is inside an AoI of a congested road (see the yellow rectangle that represents the AoI) and will pass through the congested road (see the red arrow that represents the original route of the vehicle). Hence, an alternative route is calculated and sent to the vehicle to avoid this congested road (see the green arrow that represents the new route of the vehicle).

C. Re-routing and traffic balancing

One of the main concerns of re-routing vehicles is how to decrease traffic congestion at a specific road and avoid new congestion in the near future at other roads, given changes in the traffic pattern of the former road. That is, how to avoid moving the congestion problem from one road to another. CHIMERA implements a clever strategy to overcome this problem.

First, our re-routing algorithm works under the boundaries imposed by the RSU coverage. That is, when choosing an alternative route, CHIMERA uses as starting point the vehicle's

Algorithm 1 CHIMERA algorithm.

▷ **Input:**

1: msg sent from vehicle v with vehicle' infomation (position, speed, current path, destination)
2: $\nu \leftarrow$ Set of the vehicles in the network
3: $G \leftarrow$ Graph created by each RSU

Action:

4: **for** $e_i \in G$ **do**
5:     $s_i \leftarrow$ getAverageSpeed(ν)
6:     $n_i \leftarrow$ getDensity(ν)
7:     updateWeight(s_i, n_i, e_i)
8: **end for**
9: **if** isClassificationTime() **then**
10:     $C \leftarrow$ classifyRoads()
11:     **for** $v_j \in \nu$ **do**
12:         $r_j \leftarrow v_j$.getRoute()
13:         **for** $e_i \in r_j$ **do**
14:             **if** $e_i \in C$ **then**
15:                 **if** v_j inside AoI of e_i **then**
16:                     $source \leftarrow v_j$.currentPosition()
17:                     $lastEdge \leftarrow r_j$getLastEdge()
18:                     $kPaths \leftarrow$ kShortestPaths($source, lastEdge$)
19:                     $r_{new} \leftarrow$ boltzmann($kPaths$)
20:                     **if** lastEdge != r_j.getDestination() **then**
21:                         $remaningEdges \leftarrow r_j$.split($lastEdge$)
22:                         r_{new}.add($remaningEdges$)
23:                     **end if**
24:                     sendRoute(v_j, r_{new})
25:                 **end if**
26:             **end if**
27:         **end for**
28:     **end for**
29: **end if**

current position and as ending point, the last edge in the vehicle's route that is inside the RSU coverage (Algorithm 1 – Lines 16 and 17). Therefore, route changes are localized in the coverage of each RSU. Second, CHIMERA calculates k alternative rotes by using the K-Shortest Path algorithm based on road weights (Algorithm 1 Line 18). Furthermore, using the Boltzmann probability distribution [10], CHIMERA may assign different paths to different vehicles, thus performing a load balance across alternative paths (Algorithm 1 – Line 19). It is worth noticing that, when re-routing a vehicle, if the final destination of the vehicle (Line 20) is an edge outside the coverage of the RSU performing the re-routing procedure, then the vehicle's route is divided into two parts. The first one is the new route assigned by the RSU, which comprises all edges in the vehicle's route that is inside the RSU coverage. The second part is comprised of the remaining edges in the original route that is outside the RSU coverage (Algorithm 1 – Lines 21 and 22). Finally, these two parts are concatenated and this new route is sent to the vehicle (Algorithm 1 – Line 24).

IV. Performance Evaluation

This section describes the methodology employed in our study and the main results. Section IV-A discusses the simulation tools and the scenario used in our assessment. Section IV-B shows the simulation results. First, we compare CHIMERA to solutions found in the literature. Second, we analyze CHIMERA performance according to several parameters: density, AoI size, traveled time, average speed and stopped time.

A. Methodology

We use the well-known SUMO (*Simulator for Urban MObility*) [3], version 0.17.0, to perform the evaluation. To manage the traffic, we use TraCI (Traffic Control Interface) [18]. TraCI allows traffic management at run-time during the simulation. Moreover, it provides access and control of simulation objects, changing their behaviour on-the-fly. To calculate CO_2 emissions and fuel consumption, we use the EMIT model [6] implemented in SUMO. EMIT is a simple statistical model from HBEFA[2] formula, which computes instant CO_2 emissions and fuel consumption based on acceleration and vehicle speed.

It should be stressed that we use a realistic scenario, taken from the OpenStreetMap[3] and chose the Manhattan map of 5 km^2 in New York, United States. Vehicles travel at random along the street to ensure close to real overtaking, and their speeds range from from 0 kilometres per hour to the maximum speed allowed on the road. The conditions of traffic are 250 , 500 , 1000 and 1500 vehicles/km^2. This large amount of vehicle causes congestion on the streets. First, we simulate CHIMERA under different traffic conditions by modifying the vehicle density. Second, we analyze CHIMERA according to two main parameters: the time interval t, and AoI size.

Table I shows the simulation parameters and the associated values used in our assessment. For the sake of simplicity, we performed the evaluation and show the results with the k-NN algorithm using k=5. Five was the value that reached the best performance results in our tests. Finally, for every analysis, the results represent the mean of 33 replications with a confidence interval of 95%.

Table I. SIMULATION PARAMETERS

Parameters	Values
Scenario size	5 km^2
Classification interval (CI)	15, 30, 60, 120 s
Area of interest (AoI)	300, 600, 900, 1200 m
Densities	250, 500, 1000, 1500 v/km^2
k-paths	5
# Nearest neighbours	5

We compare CHIMERA with the original traffic mobility trace (OVMT), i.e., no vehicle re-routing, and three literature solutions: DSP and RkSP, which were proposed in [12], and the solution proposed by Brennand et al. [5]. In these evaluations, the following parameters are evaluated: *i)* average speed – average speed of all vehicles; *ii)* average travel time – average time it takes for all vehicles to reach their final destination; *iii)* stopped time – average time that all vehicles get stuck in traffic jams; *iv)* travel distance – average distance that all vehicles traveled; *v)* fuel consumption – average of the fuel consumption for all vehicles; and *vi)* CO_2 emission – average of the CO_2 emitted by all vehicles.

B. CHIMERA versus literature solutions

First, we analyze the density and average speed with respect to time. In a congested scenario, the vehicle density per km^2 tends to be high and the average speed very low. Such behavior is shown in Figure 2. Second, we show that CHIMERA has a better performance than other solutions with respect to the stopped time and average speed, as shown in Figure 3. Due to space constraints, we shall not show the tests performed to adjust the parameters: routing intervals and the k value for the k-shortest path calculation. The DSP, RkSP and with routing proposed in [5] need a pre-configuration of the routing interval. We performed tests using different routing intervals for these solutions and the best routing interval obtained was 300 seconds. Another parameter to adjust is the k parameter for the k-shortest path algorithm. CHIMERA, RkSP and with routing solution introduced in [5], have a parameter k that is the number of the possible paths. Therefore, we simulated three different values for k and for all solutions, $k = 5$ reached the best results.

Figure 2 shows the correlation between speed and density for each solution during the entire simulation time with 1500 v/km^2. OVMT is the original trace and it shows how the traffic is congested, as can be seen during almost the entire simulation. The average speed is lower than 10 km/h. Moreover, the simulation lasts approximately 55 minutes and the average speed becomes greater than the density only at approximately 35 minutes. The DSP results show how efficient the periodically re-routing of vehicles is, as can be seen the average speed increase when compared to OVMT and decrease the simulation time to approximately 45 minutes. However, in DSP, when vehicles are re-routed, the best route is selected for each vehicle. Therefore, with this simple re-routing mechanism, other congestions can start at other locations in the network, however it will be detect only in the next re-routing phase. On the other hand, the RkSP solution addresses this problem, as can be seen with the increase of the average speed when compared to DSP.

In RkSP, k possible routes are calculated for each vehicle and a route is randomly chose, thus balancing the traffic and decreasing the probability of creating a new congestion in other location. However, in RkSP, routes with greater distances can be selected, thus increasing the travel time, as can be observed when the simulation time reaches 60 minutes. Differently from RkSP, which selects a route randomly, the solution presented in [5] selects a new route using a probabilistic method. The impact of this selection can be seen in the average speed, which behaves approximately like RkSP and the simulation time decreases when compared to RkSP. Finally, our solution shows the efficiency of the mechanism to detect and control the traffic jam in real time. The simulation time was reduced by approximately 50% when compared to OVMT and the average speed is greater than all other solutions, reaching up to 25 km/h. These results show the potential of our solution.

Figure 3 shows the results of all evaluated metrics for all solutions. As can be seen, OVMT has an average travel distance of 3.52 km and an average travel time of 15.79 minutes. However, vehicles spend approximately 49% of the travel time in the traffic jam (See Figure 3(c)). On the other hand, DSP decreases the average travel time in approximately 6% when compared to OVMT. This reduction is due to the periodically re-routing of all vehicles. However, this re-routing increases the average travel distance in 25% (See Figure 3(d)). Furthermore, DSP re-routes all vehicles using a shortest path algorithm. Therefore, many vehicles can be re-routed through the same route, thus creating traffic jam in other areas of the network. In fact, vehicles spend 39% of the entire travel time in a traffic jam when DSP is used (See Figure 3(c)). Differently from DSP, RkSP and the with routing solution presented in [5], distribute the traffic by selecting a route in a set of k possible routes, thus avoiding the probability of creating congestion in another area. However, RkSP increases the travel time and the travel distance by 2% and 40%, respectively, when compared to OVMT. Due to the random selection of routes, routes with

[2]http://www.hbefa.net
[3]http://www.openstreetmap.org/

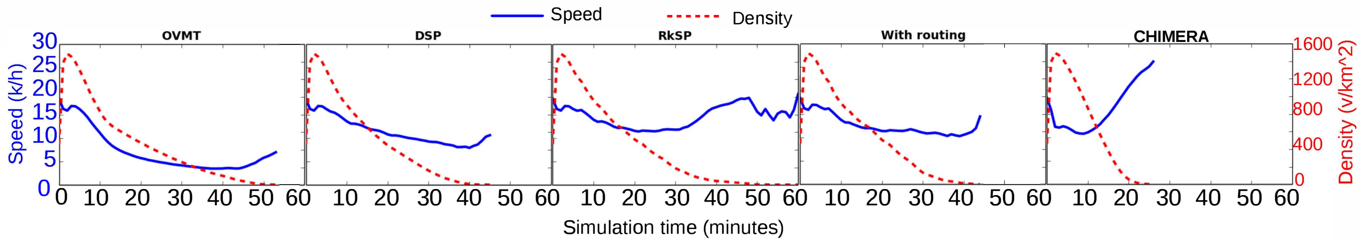
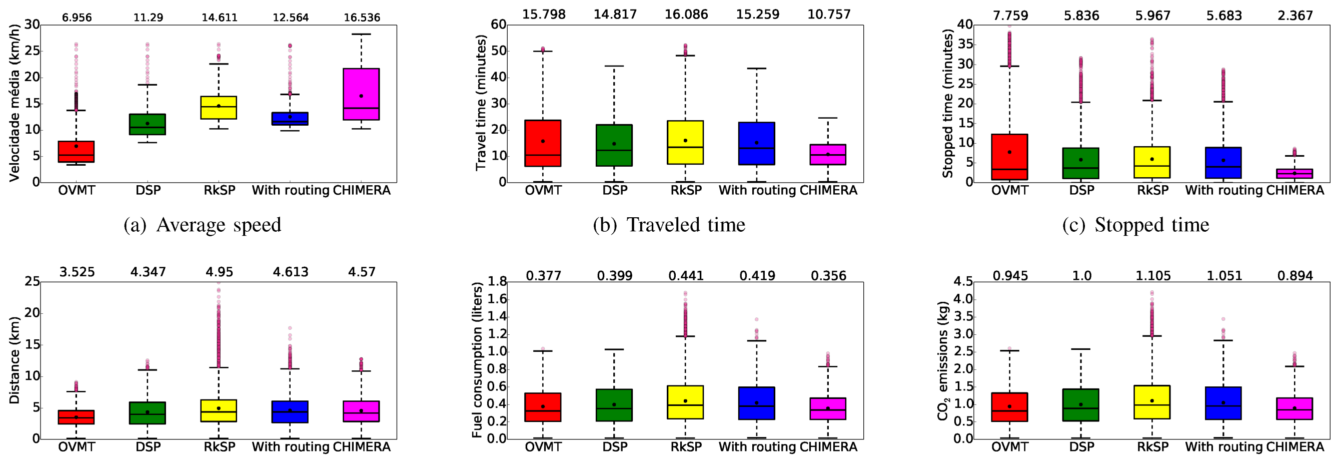


Figure 2. Correlation bettween speed and density



(a) Average speed (b) Traveled time (c) Stopped time

(d) Traveled distance (e) Fuel consumed (f) CO_2 emitted

Figure 3. Evaluation of the mechanism versus literature solutions.

greater distances can be selected, increasing the travel time and the travel distance. Moreover, vehicles spend 36% of the travel time in the traffic jam.

To avoid the issues associated with random route selection, the solution presented in [5] uses a probabilistic method to select a new route and balance the traffic. Such solution decreases the average travel time in 3% and increases the average travel distance in 30% when compared to OVMT. However, like in RkSP, vehicles spend 36% of the travel time in a traffic jam. Finally, the results of CHIMERA show how efficient the real-time congestion detection and re-routing mechanisms are. In CHIMERA, as soon as a congestion occurs, it detects and controls it, differently from the other solutions that only detect a traffic jam during the next re-routing phase. As a consequence, CHIMERA decreases the average travel time in 31% (See Figure 3(b)), the average stopped time in 70% (See Figure 3(c)) when compared to OVMT. Moreover, the traffic efficiency of CHIMERA can be observed in average speed (see Figure 3(a)) which is almost three times larger than the OVMT.

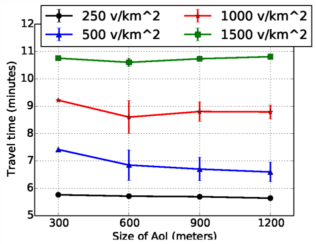
The results of the fuel consumption are directly related to the travel time, stopped time and travel distance. Figure 3(e) shows the results of the fuel consumption. As can be seen, OVMT has an average fuel consumption of 0.37 liters. DSP, RkSP and the with routing solution presented in [5] increase the fuel consumption by 5%, 16% and 11%, respectively, when compared to OVMT. Such increase is due to the greater average travel distance for these solutions. However, CHIMERA decreases the fuel consumption in approximately 6% due to the lower travel distance and lower time that vehicles spend in the traffic jam. Finally, Figure 3(f) shows the CO_2 emissions results. As expected, the CO_2 emissions are related to the fuel consumption. Therefore, the DSP, RkSP and the with routing solution presented in [5] increase the CO_2 emissions in 6%, 17% and 11%, respectively, when compared to OVMT. On the other hand, CHIMERA decreases de CO_2 emissions in approximately 6%.

C. Impact of AoI

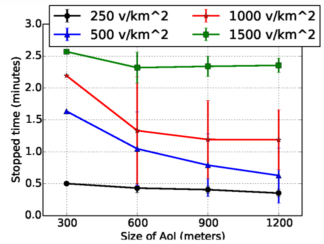
In this section, we analyze the impact the size of the AoI has on CHIMERA. As shown in Figures (4a, 4b e 4c), the denser the traffic, the greater the travel time (See Figure 4(a)) and stopped time (see Figure 4(b)) and lower the average speed (see Figure 4(c)). However, the travel time remains constant for all AoI sizes. On the other hand, as the size of the AoI increases the stopped time decreases under sparse scenarios, because more vehicles are re-routed and they have more free-flow roads as options to be re-routed. Under dense scenarios, the stopped time remains constant, because the entire scenario is congested and regardless of the traffic balancing performed by the re-routing algorithm, many roads are still congested. Thus, the size of AoI does not impact the traffic efficiency of our solution, as can be seen the average speed increases as the size of AoI increases for sparse scenarios. However, as the scenarios become denser as the size of AoI increases, the average speed decreases.

D. Impact of classification interval

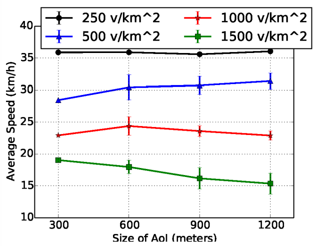
This section analyzes how the classification time impacts in our solution. As shown in Figures (4d, 4e e 4f), the



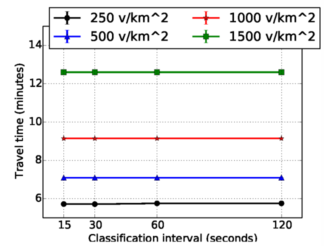
(a) Traveled time – *AoI*



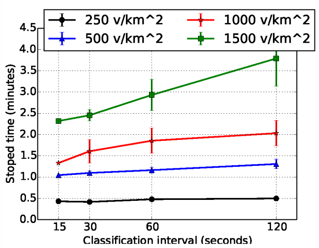
(b) Stopped time – *AoI*



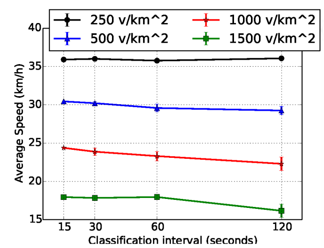
(c) Average Speed – *AoI*



(d) Traveled time – *CI*



(e) Stopped time – *CI*



(f) Average Speed – *CI*

Figure 4. Simulation results

travel time remains constant for every density regardless of the classification interval. However as the classification time increases the stopped time increases significantly to dense scenario (see Figure 4(e)). This increase in stopped time is due to the fact that the greater the classification interval is, the more vehicles enter in the congestion before the congestion is detected. As a consequence the average velocity decreases, as can be seen in Figure 4(f).Thus, the classification interval does not have a significant impact in the traffic efficiency of our solution, as can be seen the average speed decreases very little as classification interval increases.

V. Conclusion and Future Work

We proposed CHIMERA, a novel solution to detect and control traffic jam in urban centers. The proposed solution aims to reduce the traveled time, stopped time and traveled distance. Simulation results show the effectiveness of CHIMERA. When compared with an original vehicular mobility trace, CHIMERA reduces the average trip time in approximately 31%, the average stopped time in 70% and the average traveled distance in 8%. As future work, we intend to analyze the performance of our proposed solution in more realistic scenarios, using traces of real environments. In addition, we intend to extend our solution to design a cooperative re-routing algorithm.

Acknowledgments

The authors would like to thank the grant 2015/07538-1, Sao Paulo Research Foundation (FAPESP) for the financial support.

This work also has received funding from the European Union's Horizon 2020 for research, technological development, and demonstration under grant agreement no. 688941 (FUTEBOL), as well from the Brazilian Ministry of Science, Technology and Innovation (MCTI) through RNP and CTIC.

References

[1] G. Araujo, M. Queiroz, F. Duarte-Figueiredo, A. Tostes, and A. Loureiro. Cartim: A proposal toward identification and minimization of vehicular traffic congestion for vanet. In *Computers and Communication (ISCC), 2014 IEEE Symposium on*, pages 1–6, June 2014.

[2] R. Bauza and J. Gozalvez. Traffic congestion detection in large-scale scenarios using vehicle-to-vehicle communications. *Journal of Network and Computer Applications*, 36(5):1295 – 1307, 2013.

[3] M. Behrisch, L. Bieker, J. Erdmann, and D. Krajzewicz. SUMO - Simulation of Urban MObility: An Overview. In *International Conference on Advances in System Simulation*, pages 63–68, 2011.

[4] T. R. Board. Hcm 2010 - highway capacity manual, 2000.

[5] C. Brennad, A. Mariano, G. Maia, A. Boukerche, A. Loureiro, and L. Villas. An intelligent transportation system for detection and control of congested roads in urban centers. In *Computers and Communications (ISCC), 2015 IEEE Symposium on*, July 2015.

[6] A. Cappiello, I. Chabini, E. Nam, A. Lue, and M. Abou Zeid. A statistical model of vehicle emissions and fuel consumption. In *Intelligent Transportation Systems, 2002. Proceedings. The IEEE 5th International Conference on*, pages 801–809, 2002.

[7] F. D. da Cunha, L. Villas, A. Boukerche, G. Maia, A. C. Viana, R. A. F. Mini, and A. A. F. Loureiro. Data communication in vanets: Protocols, applications and challenges. *Ad Hoc Networks*, 44:90–103, 2016.

[8] R. Doolan and G.-M. Muntean. Vanet-enabled eco-friendly road characteristics-aware routing for vehicular traffic. In *Vehicular Technology Conference (VTC Spring), 2013 IEEE 77th*, pages 1–5, 2013.

[9] G. Karagiannis, O. Altintas, E. Ekici, G. Heijenk, B. Jarupan, K. Lin, and T. Weil. Vehicular networking: A survey and tutorial on requirements, architectures, challenges, standards and solutions. *Communications Surveys Tutorials, IEEE*, 13(4):584–616, Fourth 2011.

[10] S. Kirkpatrick, C. D. Gelatt, M. P. Vecchi, et al. Optimization by simmulated annealing. *science*, 220(4598):671–680, 1983.

[11] D. T. Larose. *k-Nearest Neighbor Algorithm*, pages 90–106. John Wiley and Sons, Inc., 2005.

[12] J. Pan, M. Khan, I. Sandu Popa, K. Zeitouni, and C. Borcea. Proactive vehicle re-routing strategies for congestion avoidance. In *Distributed Computing in Sensor Systems (DCOSS), 2012 IEEE 8th International Conference on*, pages 265–272, May 2012.

[13] A. M. Souza, G. Maia, and L. A. Villas. Add: A data dissemination solution for highly dynamic highway environments. In *Network Computing and Applications (NCA), 2014 IEEE 13th International Symposium on*, pages 17–23, Aug 2014.

[14] A. M. d. Souza, A. Boukerche, G. Maia, R. I. Meneguette, A. A. Loureiro, and L. A. Villas. Decreasing greenhouse emissions through an intelligent traffic information system based on inter-vehicle communication. In *Proceedings of the 12th ACM International Symposium on Mobility Management and Wireless Access*, pages 91–98, 2014.

[15] L. A. Villas, A. Boukerche, R. B. Araujo, A. A. F. Loureiro, and J. Ueyama. Network partition-aware geographical data dissemination. In *2013 IEEE International Conference on Communications (ICC)*, pages 1439–1443, June 2013.

[16] L. A. Villas, A. Boukerche, G. Maia, R. W. Pazzi, and A. A. Loureiro. Drive: An efficient and robust data dissemination protocol for highway and urban vehicular ad hoc networks. *Computer Networks*, 75, Part A(0):381–394, 2014.

[17] M. Wang, H. Shan, R. Lu, R. Zhang, X. Shen, and F. Bai. Real-time path planning based on hybrid-vanet-enhanced transportation system. *Vehicular Technology, IEEE Transactions on*, May 2015.

[18] A. Wegener, M. Piórkowski, M. Raya, H. Hellbrück, S. Fischer, and J.-P. Hubaux. Traci: an interface for coupling road traffic and network simulators. In *Proceedings of the 11th communications and networking simulation symposium*, pages 155–163. ACM, 2008.