2016 IEEE Symposium on Computers and Communication (ISCC)

FOX: A Traffic Management System of Computer-Based Vehicles FOG

Celso A. R. L. Brennand*§, Felipe Domingos da Cunha†, Guilherme Maia†,
Eduardo Cerqueira‡, Antonio A.F. Loureiro† and Leandro A. Villas*
*Institute of Computing, University of Campinas, Brazil
†Federal University of Minas Gerais, Brazil
‡Institute of Computing, Federal University of Pará, Brazil
§UAST, Federal Rural University of Pernambuco, Brazil
Email: celso@lrc.ic.unicamp.br, {fdcunha, jgmm, loureiro}@dcc.ufmg.br, cerqueira@ufpa.br and leandro@ic.unicamp.br

***Abstract*—Traffic congestion causes drivers' frustration and costs billions of dollars annually in lost time and fuel consumption. In order to overcome such issues, this paper presents a mechanism for Intelligent Transport Systems named FOX (Fast Offset XPath), which aims to detect and manage traffic congestion in Vehicular Ad hoc Networks. FOX is implemented in a FOG computing environment, taking advantage of the aspects inherent to this platform, such as scalability, low latency, the importance of geographical location and network conditions. The focus is to reduce the time to process, reroute and notify vehicles. Simulation results show that the proposed mechanism can reduce the average trip time, CO_2 emissions and fuel consumption. In particular, the average trip time was decreased approximately in 32%, the average fuel consumption in 14% and the stop time in 59%.**

I. Introduction

Traffic congestion is a recurrent problem in major urban centers and directly affects the economy. In addition, it also impacts the productivity of the society as a whole since congestion increases the time that people need to move from one point to another. It also negatively interferes in the environment, as a result of the increase in CO emissions. The study conducted in [1] shows that the monetary cost due to congestion in the city of São Paulo in the year 2012 was in the order of US$ 80 billion. Around the world, equivalent cities have similar numbers. Of this total, 85% is associated with time wasted in traffic, 13% is related to the increase in fuel consumption and 2% corresponds to the increase in CO emissions.

The Intelligent Transport System (ITS) attempts to prevent congestion and to improve the efficiency of transport systems. ITSs use advances in technology in the areas of processing, sensing and communication in order to monitor the traffic conditions in a particular region, to manage and to reduce congestion. In addition, they can be employed to provide information and entertainment services to drivers and passengers, so that their journey is a more pleasant process [2], [3], [4], [5]. An important component in the design of ITSs are Vehicular Networks (VANETS). In these networks, vehicles are equipped with processors, sensors and wireless communication interfaces. The vehicles communicate with each other and with the elements belonging to the network infrastructure (RSU – Road Side Unit) or vehicle-to-infrastructure (V2I), creating a VANET.

Some research in the literature have proposed many architectures to design ITSs, ranging from centralized solutions [6] to distributed solutions [7]. In general, these solutions use both information provided by vehicles and the characteristics of the road infrastructure in order to detect and to control congestion. Thus, after detecting an area or road congested, a re-routing mechanism is applied to the vehicles to prevent them from entering the affected area, resulting in a more efficient traffic. One major challenge in the design of ITSs is the huge volume of data generated by both sensors embedded in vehicles and those deployed in a city that need to be delivered to different stakeholders. The response time is also another problem because VANETS are very dynamic [8] and system outputs must reach intended recipients while the information is still useful.

In this context, this paper presents a route management mechanism for ITS, named Fast Offset Xpath - FOX. FOX uses the FOG computing paradigm [9] to detect and minimize congestions in real time. For that, RSUs are distributed across the city to ensure the total coverage of the region, as expected in smart cities. Each RSU belongs to a FOG. The idea of the FOG paradigm is to move its resources (processing and storage) to the edge of network, making the resources available as close as possible to the end-users, which brings benefits in the design of an ITS, such as [10]: *predominant wireless access* – modern ITS systems heavily rely on wireless communications; *low latency* – some ITS data have strict time constrains, such as data for re-route systems; *wide geographical distribution* – ITS has sensors geographically distributed, however the scope of the data gathered is restricted to the location of the sensors that generated such data; *mobility* – an ITS is used to optimize the mobility of vehicles in the city, however the ITS may also leverage mobility to perform data delivery activities to different stakeholders; *scalability* – an ITS needs to be scalable due to the great number of sensors and vehicles; *extensibility* – if the city grows, the ITS also needs to grow up in order to accommodate the expansion of the area; and *real-time interaction* – re-routing systems have real-time requirements.

FOX uses FOG characteristics in order to reduce the harmful effects associated to congestion, such as long travel times, as well as to reduce the fuel consumption and CO_2 emissions. As every FOG in the proposed route management mechanism is associated to one RSU, each FOG is responsible for the control of the congestion, independently and autonomously, only in the area covered by its RSU. This decreases the network load, the computation and response time.

The remainder of this paper is structured as follows. Section II presents an overview of the literature approaches to minimize congestion in urban centers. Section III presents the FOX re-routing mechanism and its components. Section IV presents the performance evaluation of our proposal, along with the methodology used and the results. Finally, section V discusses the conclusions and future work.

II. Related Work

This work presents FOX, a real time mechanism to optimize the flow and the movement of vehicles in urban centers. FOX aims to minimize the congestion of vehicles and, consequently, to reduce travel time, fuel consumption and CO_2 emissions. The problem of vehicle congestion has been explored in the literature for years, and some solutions have been proposed to optimize the flow of vehicles in cities [11], [12], [6], [7]. Among these, the solutions [6] and [7] are similar to our proposal, as discussed below.

In [6], the authors proposed a system for predicting, detecting and controlling congestion. The system uses a centralized approach to gather information about vehicular traffic in real-time (position, speed and direction). The information collected is used to detect or to predict a traffic jam. After the detection or prediction of a congestion, the system has a congestion control mechanism, where vehicles approaching congested areas are re-routed to alternative paths. The system employs three strategies for re-routing the vehicles. The first is called Dynamic Shortest Path (DSP), which re-routes vehicles through the shortest alternative path. The second strategy is the Random k Shortest Paths (RkSP), which selects k shortest path routes and assign, at random, one of them to the vehicle. Finally, the strategy Entropy Balanced k Shortest Paths (EbkSP) extends the RkSP strategy by using a smart mechanism for route selection, which considers the impact that each of the k paths has on the future density of the road.

In [7], the authors propose EcoTrec, a distributed algorithm for re-routing vehicles based on V2V communications. EcoTrec makes use of features of the highway network and traffic conditions to enhance traffic efficiency and to reduce CO_2 emissions and fuel consumption. In this algorithm, each vehicle periodically disseminates its route and its fuel consumption. Then, this information is used to compute the best routes for each vehicle. If necessary, new routes are suggested for vehicles with lower fuel consumption and CO_2 emissions.

It is important to highlight that these solutions lacks some features compared to FOX. In the work of [6], all the re-routing is done centrally, by a single RSU responsible for communicating and calculating alternative routes for all vehicles. This typically causes a high traffic band in the network and longer time for the calculation of alternative paths when the road network is large. On the other hand, in the work of [7], re-routing is done in a distributed way, where each vehicle calculates its own alternative route. However, since vehicles do not have full knowledge of the whole network, only their local neighborhood, many vehicles may calculate the same alternative routes, resulting in new traffic jams. Furthermore, a large number of messages is exchanged among vehicles, thus compromising system scalability.

With this issues in mind, our solution proposes the use of a distributed FOG approach relying on several RSUs. In this way, the time for the calculation of alternative routes becomes much smaller when compared to [6], for instance, because each RSU is responsible only for the re-routing of vehicles located in its coverage area. Moreover, by using a distributed re-routing approach, the problems found in [7] are eliminated because each fog has knowledge of a region of the map, thus avoiding the assigment of the same route to several vehicles in its region. Due to the amount of generated data and the high demand for computational resources of ITSs, some solutions attempted to integrate VANETs with cloud computing and its variations [13], [14], [15], [16]. The main disadvantage of such approaches is that the cloud paradigm demands a high and constant network communication band. This occurs because cloud data centers are typically far away from the end-users, thus requiring the transport of large amounts of data in the network. One characteristic of ITSs is that the gathered data is spatially correlated [17] and relevant to certain geographical regions, therefore sending all the data to a centralized place is unnecessary. On the other hand, aggregating and processing correlated data locally is more effective. Due to these factors, we propose an architecture based on FOG computing [10].

III. FOX: Fast Offset Xpath

FOX: Fast Offset Xpath is a route management mechanism where RSUs and sensors form computing FOGs, putting both computing and data closer to where they are required. In this way, FOX geographically distribute the processes of data collection, processing and system response, making them closer to vehicles, which decreases the costs of communication on the whole network and the system response time.

FOX divides the city into regions, where each region corresponds to a computing FOG, which are independent of each other. A FOG is composed of a single RSU equipped with processing, storage and wireless communication, sensors and vehicles. The size of the region defined by a FOG corresponds to the communication coverage of its RSU. In this way, each FOG is responsible for detecting and controlling congestion within its radio coverage. Notice, however, that a FOG possesses direct knowledge only about the part of the city (roads, topology, level of congestion, etc) that encompasses the communication coverage of its RSU. For each FOG, there is also an Area of Interest (AoI). AoI is the region of the map where the FOX mechanism has knowledge (roads and its conditions) to make re-routing of a vehicle that is inside of its RSU coverage. The AoI has at least the size of the RSU coverage. In order to retrieve knowledge about roads that are under the responsibility of another FOG, one FOG can communicate directly with other FOG and request the desired information. Each FOG can independently and dynamically set the AoI size according to the time of day and the density of its region, thus improving the outcome of the solution and the load balancing.

Figure 1 illustrates the FOX management mechanism. For instance, in the upper right we can see a FOG and its corresponding RSU in blue, its communication radius in blue and its AoI in green. For the sake of simplicity, in this paper, we will use the area of the FOG as a circle centered at the position of its correspondent RSU and the radius of the circle equal to the communication radius of the RSU. In addition, the proposed system is composed of a congestion control mechanism, which makes the re-routing of all vehicles periodically in order to avoid traffic through congested areas. This control system is also part of the functionality of the FOG. FOX can be configured with parameters such as routing interval, number of alternatives routes K and AoI according to certain conditions of the scenario, such as time of day (rush hours or quiet hours) and density, but it is out of the scope of this paper.

Other details of the mechanism are explained in the following subsections. The system can be divided into three stages,

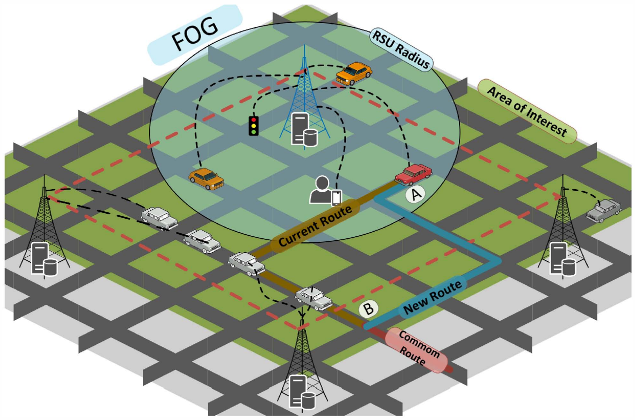


Fig. 1. FOX: Fast Offset Xpath

namely: (i) Distribution of the RSU (FOG); (ii) Collecting and Transmission of Data to the RSU; (iii) Detection and Control of Congestion.

A. *Distribution of the RSUs*

The proposed mechanism uses some RSUs distributed in the environment to accomplish full coverage of the map. The distribution of the RSUs on a map can be made in a variety of ways. For example, making a relationship between coverage of the RSU and either the density of vehicles in the region or the number of roads. Then, it is possible to evaluate the trade-off between the quality of the response given by the solution and response time given to end-users. For the sake of simplicity, this work uses an homogeneous distribution based on the communication radius of the RSUs and the map dimensions.

The communication radius of the RSUs has a direct impact on the monetary cost to perform the RSUs deployment. For instance, the smaller the communication radius is, the greater is the number of RSUs to ensure the coverage of the entire map. With more RSUs, communication will be restricted to smaller areas, thus decreasing the system overhead.

B. *Collecting and Transmission of Data to the RSU*

The data collection is made by the RSU of each FOG in order to acquire information concerning the events that occurred within their communication radius and also to understand the behavior of vehicles on the roads in its area. Thus, vehicles periodically send information (position, speed, current route, direction and the time spent to move through each road of its route) to the RSU. The information provided by the vehicles is sent to the RSU, since both are equipped with communication capabilities, such as IEEE 802.11p [18] or Long Term Evolution (LTE) technology [19]. After the data collection, there is a data processing phase in each RSU. Finally, the information generated by the RSU can be transmitted to the other RSUs intersecting the AoI of the FOG (see Figure 1).

The size of the AoI influences the performance of the solution in two ways: first in its effectiveness, since the greater the area the more knowledge about the map the RSU will have; and in its efficiency, because with a smaller area, the faster is the processing of the congestion control system. The results presented in Section IV-C show the impact of the AoI size of the FOGs in the proposed system.

C. *Detection and Control of Congestion*

With the information reported by vehicles and other RSUs, each FOG identifies the characteristics of the region of its AoI. Therefore, in possession of these characteristics, each FOG creates a graph $G = (V, E)$, so that V is the set of crossovers within the range of its AoI (representing the vertices) and the set of roads connecting the intersections E (representing the edges). Each road (edge) of the graph G has a weight defined as one minus the ratio between the average speed at which vehicles travel on the road and the maximum allowed speed on it. Notice that the weight is inversely proportional to the speed at which vehicles travel on the road, so that the closer the speed of vehicles in relation to the maximum allowed speed is, the lower the weight of the road (edge) is.

The re-routing of vehicles is performed periodically, so that each FOG is responsible for re-routing only vehicles within the communication radius of its RSU. Therefore, vehicles are re-routed in the range of its AoI (green area in Figure 1). A handoff between the vehicle and the RSU is not necessary, because vehicles periodically send information to the FOG. So, if the FOG is in the routing phase, the vehicles that send information at this time will receive a new route. For that, initially, the FOG performs the congestion detection using the weights of the roads in order to identify which roads are under severe load. Thereafter, the FOG performs the congestion control by re-routing vehicles to alternative roads, thus decreasing the load on the congested roads. As it can be seen in Figure 1, the re-routing of vehicles (e.g. red car) is made considering its current position (point A) until the last road in its current route that is within the AoI of the FOG (point B). In our system, the metric used to perform the re-routing is the weight previously described.

During the re-routing step, for each vehicle, the system calculates a set of k alternative shortest paths as possible routes the vehicle can take, where k can be easily adjusted to be better adapted to the behavior of the network. From these alternative routes, a route is probabilistically selected using the Boltzmann probability distribution [20]. This way, the system avoids the problem in which vehicles with similar routes are re-routed to the same alternative routes, thus creating new congestion points (one of the drawbacks of the solution proposed in [7]). After the selection of the alternative route, the system evaluates whether the size of this new route is smaller than the value of the current route plus a percentage set by the user. Several percentage values have been evaluated in Section IV-C in order to assess their impact on the overall system performance.

Algorithm 1 describes the main steps of the re-routing process. The algorithm has as input the set of vehicles within the range of each RSU (represented in the algorithm as N), the graph of the AoI created with its own congestion characteristics (represented in the algorithm as G), the factor that defines the maximum size of the new route (represented in the algorithm as ACC) and a value K that represents the maximum number of alternative paths that must be calculated.

The choice of the route between the K smallest routes is made in such a way that the load of vehicles on the roads is balanced. Each route has a weight that is calculated as the sum of the weights of all the roads it contains. As the routes with lower weight are most requested, our system uses a set of

Algorithm 1: Detection and control of congestions

```
  Input  : N // Set of vehicles within the radius of each RSU
1 G // Graph created by each FOG (area of interest)
2 K // number of alternate paths
3 ACC // percentage of size of the new route
4 foreach v ∈ N do
      // sets of edges that make up the route of vehicles v
5     route ← v.getRoute();
      // returns the current edge of vehicle v to the last edge of the path
         of vehicle v contained in graph G
6     source ← v.getPosition();
7     lastEdge ← G.getLastEdge(route);
      // calculates the k shortest paths of the source point to lastEdge
         for vehicle v
8     alternativeRoutes ←
      G.getKShortestPaths(source, lastEdge, K);
      // selects a path from the set of alternate paths, concatenates the
         remainder of the old path to the new path and assigns the new path
         to vehicle v
9     newRoute ← boltzmann(alternativeRoutes, G);
10    interestEdges, remaningEdges ←
      route.split(lastEdge);
11    if routeSizeRelation(newRoute, interestEdges, ACC)
      then
12        if lastEdge! = route.getDestination() then
              // returns the remainder of the route of the vehicle v from
                 the lastEdge
13            newRoute.add(remaningEdges);
14            v.setRoute(newRoute);
15        end
16        else
17            v.setRoute(newRoute);
18        end
19    end
20 end
```

smallest routes R_j in order to decrease the probability of always choosing the same route, thus balancing the load between the roads. Decision rules to choose the new route are found in [21].

After the route is chosen, it is verified whether the route can be accepted. This is done by checking if the new route size plus a percentage factor ACC is shorter than the size of the old route (line 11). If false, the current route is not modified and this step ends. If true, the system checks whether the last edge of the calculated alternative route is the final destination of vehicle v (Line 12). If this condition is not satisfied, the new alternative route is concatenated to the remainder of the original route that lies outside the AoI of the FOG that performed the re-routing of v (Lines 13–14).

IV. Performance Evaluation

This section presents the methodology used to evaluate the proposed solution, as well as the main results of FOX. Section IV-A discusses the simulation tools and the simulation scenario used in our assessment. Section IV-B presents the results obtained by the FOX mechanism and the comparison with the traditional approach (no routing - NR), where vehicles have no interaction with any system that provides re-routing, and two related approaches: DSP and RKSP [6]. In Section IV-C, we analyze FOX's performance by studying two system parameters: the size of the AoI and re-routing time interval.

A. Methodology

The simulations were performed using the network simulator OMNeT++ 4.3[1]. For the simulation of traffic and mobility of vehicles, we employed the SUMO simulator (Simulation of Urban MObility)[2]. For the calculation of the CO_2 emissions and fuel consumption, we used the EMIT model integrated in SUMO. EMIT is a simple statistical model for instant emissions of CO_2 and fuel consumption based on the acceleration and speed of vehicles, which is derived from the formula HBEFA[3] – Handbook Emission Factors for Road Transport. For the simulation and evaluation of our system, 33 simulation repetitions were carried out and the results present the values with a confidence interval of 95%. We used a realistic scenario obtained from OpenStreetMap[4]. The map chosen corresponds to the Manhattan area in New York City, United States, with an area of $5km^2$. Vehicles travel with randomly chosen routes and the vehicle density in the area ranges from 1000 to 1500 vehicles/km^2 during the simulation.

As described earlier, the distribution of the RSU is homogeneous and it is based on the size of the communication radius and the map dimensions, so that the larger the communication radius is, the smaller the amount of RSU used to accomplish full coverage of the map is. In our assessment, different configurations of the radius size of the AoI have been tested in the FOX to assess the impact of these changes.

When the simulation reaches a steady state, the mechanism FOX starts the re-routing of vehicles in order to minimize congestion already in place. Thus, different parameters values are varied to analyze the best combination among them.

The *re-routing intervals* were tested, being them 150, 300 and 600 seconds. For the re-routing of vehicles, our algorithm has a parameter $K - routes$ that determines the number of alternative paths provided, hence we evaluated the values 3 and 5. For the *route size factor*, which determines how much bigger (in percent) the new route can be when compared to the old one, we evaluated the values 25%, 50%, 75% and 100% (always accept). It was also varied the amount of vehicles that accepts the alternative route (ACC) by considering 25%, 50%, 75% and 100% for simulate users that not accept the route for any other reason. The radius of the Area of Interest (AoI) for RSU, we evaluated $1000m$, $2000m$, $4000m$. FOX can set dynamically theses values such according to the day time and the region density.

Finally, for the validation of our system, the following metrics were evaluated: *Traveled Time:* the average travel time from the starting point to the destination of all vehicles; *Stopped time:* average time spent stuck in traffic jams for all vehicles; *Average speed:* average speed of all vehicles; *Traveled distance:* average distance that all vehicles traveled; *Fuel consumption:* average fuel consumption of all vehicles to traverse the whole route; *CO_2 emission:* average CO_2 emissions for all vehicles during their journey.

B. FOX Mechanism versus Literature Approaches

The routing solutions FOX, DSP, RkSP [6] need to pre-configure the routing interval. Tests with different re-routing

[1]http://www.omnetpp.org
[2]http://www.sumo.dlr.de
[3]http://www.hbefa.net
[4]http://www.openstreetmap.org/

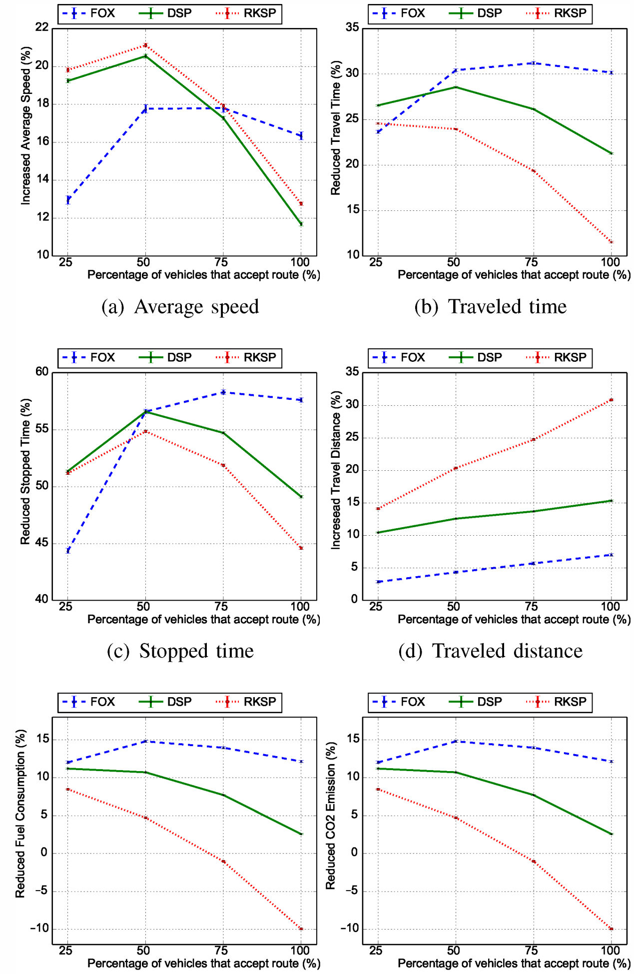
intervals were performed and the best routing interval obtained was 150 seconds. The parameter k was also studied and $k = 3$ produced the best results for the algorithms FOX and RKSP. The value $k = 5$ presented the worst results since it led to longer (worse) paths, thus directly affecting the solutions. Results only show the density of 1500 vehicles/Km^2, because such density causes severe congestion, demanding more work from the all solutions.

Figure 2 shows the percentage increase or decrease of all metrics for all protocols in relation to the baseline solution (no re-routing) when varying the *percentage of vehicles that accepts to be re-routed* (25%, 50%, 75%, 100%). As we can see in Figure 2(a), when 25% of the vehicles are re-routed, DSP and RKSP present an increase of about 20% in the average speed when compared to the baseline solution, while for FOX such increase is about 13%. This means that DSP and RKSP are handling traffic jams more efficiently than FOX under this configuration. However, when at least 50% of vehicles are re-routed, the increase of the average speed for both DSP and RKSP when compared to the baseline solution starts to decay (12% when 100% of vehicles are re-routed), while for FOX it is rather constant, at about 16%. This result shows that FOX is more efficient in handling traffic jams when a high number of vehicles are re-routed.

A similar behavior can be observed when we look to the travel time (Figure 2(b)). When 25% of vehicles are re-routed, all protocols reduce the travel time in about 25% when compared to the baseline solution. However, when more vehicles are re-routed (at least 25%), FOX is the solution that induces the highest decrease in the travel time. This result shows that in FOX, vehicles are being re-routed through alternative routes that are faster than the routes selected by DSP and RKSP. This fact is corroborated by the result shown in Figure 2(c), which shows the reduction in the time that vehicles remain stopped at traffic jams. As we can see, when 100% of the vehicles are re-routed, FOX reduces the stopped time in about 58%, while for DSP and RKSP this reduction is about 50% and 45%, respectively.

When considering the length of the alternative routes provided by the solutions (Figure 2(d)), we can see that all solutions suggest to vehicles alternative routes that are longer than the original ones. Notice, however, that in FOX, alternative routes are not 10% longer than the original ones, while in DSP and RKSP, alternative routes can be 15% and 30% longer, respectively. This results can be explained by the fact that FOX employs a parameter that controls how much longer alternative routes can be.

Finally, figures 2(e) and 2(f) show the reduction in fuel consumption and CO_2 emissions when compared to the baseline solution. As we can see, independently of the number of vehicles re-routed, FOX reduces fuel consumption and CO_2 emissions in about 10%. An interesting fact in these results is that when at least 75% of vehicles are re-routed, the fuel consumption and CO_2 emissions for DSP and RKSP actually increase. This is strictly related to the longer alternative routes provided by these solutions, as shown in the previous result. In summary, these results show that when at least 75% of vehicles are re-routed, the FOX re-routing mechanism is better suited to handle traffic jams when compared to the other solutions.



(a) Average speed (b) Traveled time

(c) Stopped time (d) Traveled distance

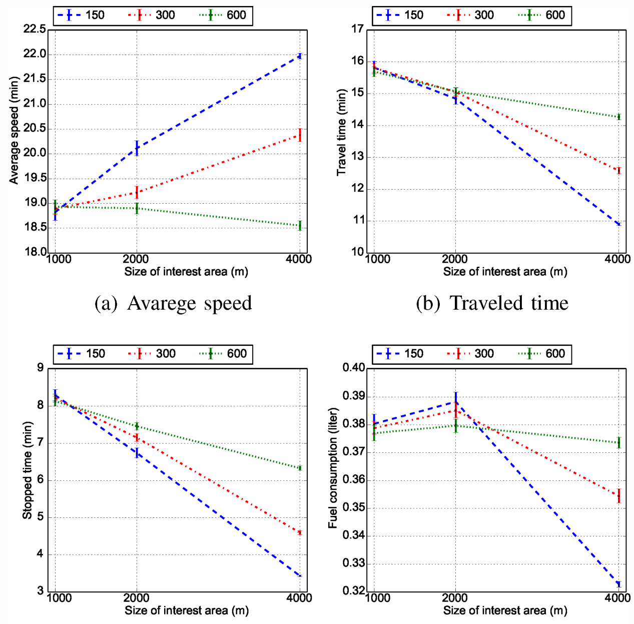
(e) Fuel consumption (f) CO2 emission

Fig. 2. Simulation results.

C. Evaluation of FOX with Different Configurations

We now vary the parameters *AoI* and *re-routing interval* for the FOX mechanism, as shown in Figure 3. As we can see, the performance of the FOX mechanism improves with the growth of the radius of the *AoI*. This occurs because a great *AoI* implies more knowledge about the map (e.g. more roads and its traffic conditions), thus it is possible to calculate a more effective alternate route.

Moreover, a shorter *routing interval* is more efficient when the *AoI* is greater. When the *AoI* is small, the use of a constant re-routing interval implies in many best local routes being chosen, however they can be far from the best global solution, as shown in Figure 3(a). Using 150 seconds for the *re-routing interval* and the *AoI* with radius of 1000 m, there was a 1% of improvement. For an *AoI* with radius 4000 m, FOX presented an improvement of about 18%. A long re-routing interval time is not effective under larger *AoI*, because the route calculated may possess a road that will become congested before the next re-routing cycle. This fact can be observed in Figure 3(d), where the fuel consumption has the best results when considering an interval of 150 seconds, leading to a reduction of about 14%.



(a) Avarege speed (b) Traveled time

(c) Stopped time (d) Fuel consumption

Fig. 3. Evaluation of the area of interest by routing on FOX.

V. Conclusion

In this work, we proposed FOX, a mechanism that employs FOG computing concepts in the design of an ITS to detect and to control congestion in urban centers. Therefore, FOX benefits from the advantages provided by the FOG paradigm, such as approximation of computer resources to end-users, which decreases the system response time and bandwidth usage. The focus of our proposed ITS solution is to reduce the travel time, fuel consumption, and CO2 emissions. Simulation results show that the proposed system significantly reduces the these metrics. For instance, FOX was able to reduce the travel time by approximately 32%, the fuel consumption by 14% and, finally, the stop time presented a reduction of about 59%.

As future work, we aim to create an architecture with this proposed mechanism by appending two new layers, the Cloud interaction and the Dynamic FOG. The Cloud will store the historical information about the scenario, processing a long-term congestion prediction and analysis of urban computing data that will be used in the prediction. The architecture will also be able to provide other ITS services, such as distribution of contents. The Dynamic FOG will be formed by a set of vehicles that are geographically close. Finally, we intend to propose a new method to classify the level of congestion of vehicles on roads and regions, using the current system information as well as data coming from the urban computation, thus providing more precision in the data to be used by the 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] M. Cintra, "A crise do trânsito em são paulo e seus custos," *GVExecutivo*, vol. 12, no. 2, pp. 58–61, Jul/Dez 2013.

[2] L. A. Villas, H. S. Ramos, A. Boukerche, D. L. Guidoni, R. B. Araujo, and A. A. Loureiro, "An efficient and robust data dissemination protocol for vehicular ad hoc networks," in *Proceedings of the 9th ACM Symposium on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks*.

[3] 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)*, June 2013, pp. 1439–1443.

[4] 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*, vol. 75, Part A, no. 0, pp. 381 – 394, 2014.

[5] 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*, vol. 44, pp. 90–103, 2016. [Online]. Available: http://dx.doi.org/10.1016/j.adhoc.2016.02.017

[6] 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*, May 2012, pp. 265–272.

[7] 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*, June 2013, pp. 1–5.

[8] W. Viriyasitavat, F. Bai, and O. K. Tonguz, "Dynamics of network connectivity in urban vehicular networks," *Selected Areas in Communications, IEEE Journal on*, vol. 29, no. 3, pp. 515–533, 2011.

[9] F. Bonomi, R. Milito, P. Natarajan, and J. Zhu, "Fog computing: A platform for internet of things and analytics," in *Big Data and Internet of Things: A Roadmap for Smart Environments*. Springer, 2014.

[10] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the internet of things," in *Proceedings of the first edition of the MCC workshop on Mobile cloud computing*. ACM, 2012, pp. 13–16.

[11] J. Fukumoto, N. Sirokane, Y. Ishikawa, T. Wada, K. Ohtsuki, and H. Okada, "Analytic method for real-time traffic problems by using contents oriented communications in vanet," in *Telecommunications, 2007. ITST '07. 7th International Conference on ITS*, 2007, pp. 1–6.

[12] R. Bauza, J. Gozalvez, and J. Sanchez-Soriano, "Road traffic congestion detection through cooperative vehicle-to-vehicle communications," in *Local Computer Networks (LCN), IEEE 35th Conference on*, Oct 2010.

[13] M. Whaiduzzaman, M. Sookhak, A. Gani, and R. Buyya, "A survey on vehicular cloud computing," *Journal of Network and Computer Applications*, vol. 40, pp. 325–344, 2014.

[14] R. Yu, Y. Zhang, S. Gjessing, W. Xia, and K. Yang, "Toward cloud-based vehicular networks with efficient resource management," *Network, IEEE*, vol. 27, no. 5, pp. 48–55, 2013.

[15] S. Olariu, T. Hristov, and G. Yan, "The next paradigm shift: from vehicular networks to vehicular clouds," *Mobile ad hoc networking: cutting edge directions. 2nd ed. John Wiley & Sons, Inc., Hoboken*, 2013.

[16] S. Bitam, A. Mellouk, and S. Zeadally, "Vanet-cloud: a generic cloud computing model for vehicular ad hoc networks," *Wireless Communications, IEEE*, vol. 22, no. 1, pp. 96–102, 2015.

[17] C. Liu, C. Chigan, and C. Gao, "Compressive sensing based data collection in vanets," in *Wireless Communications and Networking Conference (WCNC), 2013 IEEE*. IEEE, 2013, pp. 1756–1761.

[18] D. Eckhoff, C. Sommer, and F. Dressler, "On the necessity of accurate ieee 802.11p models for ivc protocol simulation," in *Vehicular Technology Conference (VTC Spring), 2012 IEEE 75th*, 2012, pp. 1–5.

[19] S. Jimaa, K. K. Chai, Y. Chen, and Y. Alfadhl, "Lte-a an overview and future research areas," in *Wireless and Mobile Computing, Networking and Communications*, Oct 2011, pp. 395–399.

[20] S. Kirkpatrick, C. D. Gelatt, M. P. Vecchi *et al.*, "Optimization by simmulated annealing," *science*, vol. 220, no. 4598, pp. 671–680, 1983.

[21] C. A. Brennand, A. M. de Souza, G. Maia, A. Boukerche, H. Ramos, A. A. Loureiro, and L. A. Villas, "An intelligent transportation system for detection and control of congested roads in urban centers," in *IEEE Symposium on Computers and Communication*, 2015, pp. 476–481.