Computer Networks 110 (2016) 118–132

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

ICARUS: Improvement of traffic Condition through an Alerting and Re-routing System

Allan M. de Souza[a,*], R.S. Yokoyama[a], Azzedine Boukerche[b], Guilherme Maia[c], Eduardo Cerqueira[d], Antonio A.F. Loureiro[c], Leandro Aparecido Villas[a]

[a] Institute of Computing, University of Campinas, Campinas 13080590 Brazil
[b] School of Electrical Engineering and Computer Science, University of Ottawa, Canada
[c] Computer Science Department, Federal University of Minas Gerais, Brazil
[d] Federal University of Para, Belem, PA, Brazil

ARTICLE INFO

Article history:
Received 12 February 2016
Revised 13 September 2016
Accepted 18 September 2016
Available online 21 September 2016

Keywords:
Congestion control
Traffic Management System
VANET
Data dissemination
Re-routing

ABSTRACT

In cities, where the number of vehicles continuously increase faster than the available infrastructure to contain them, traffic congestion is a difficult issue to deal with. This problem becomes even worse in case of accidents and affects many aspects of the modern society, including economic development, accidents, CO (Carbon monoxide) emission, trip time, and health. Several solutions for Traffic Management System (TMS) have been proposed to identify congestions and re-route the vehicles afterward. To this end, they exchange messages periodically between vehicles and central server, what can cause an overhead in the communication channel. In this scenario, it is important to identify the source of the problem and inform the drivers of new routes before the congestion takes place with, considering the limitations of vehicular communication. This work introduces ICARUS, a distributed and pro-active Traffic Management System, which receives notifications about a traffic events then it can calculates new routes, and, then, notifies drivers to follow new paths pro-actively by using inter-vehicle communications. Simulation results show the effectiveness of ICARUS in calculating new routes and disseminating them to vehicles approaching a congested area. Hence, ICARUS reduces the travel time, fuel consumption, and CO emissions of vehicles in urban environments when compared to existing approaches. In addition, ICARUS reduces the broadcast storm problem and maximizes the data dissemination capabilities with short delays and low overhead.

1. Introduction

The world population is concentrated in large cities and it has mobility as one of its most basic needs. The preferred way to accomplish mobility in modern societies is through the use of automotive vehicles [1]. As a consequence, modern cities have been suffering with the steep increase in the number of vehicles. Since the road infrastructure does not grow at the same pace as the number of vehicles, traffic congestion has become a pressing issue in the largest cities around the world. It creates a number of negative issues for the society, such as, the increase in the number of car accidents, negative effects on economic development and negative impact on the environment [2,3].

According to a U.S. Department of Transportation report, there are three main sources for congestion [4]. The first one is related to traffic-influencing events, such as incidents, working zones, and bad weather. The second one is related to traffic demand, which means fluctuations in normal traffic and special events. The last source is related to the road infrastructure features, which represent the traffic control devices and physical bottlenecks. Moreover, the report also shows that bottlenecks are responsible for 40% of the overall congestion, followed by incidents, such as car accidents with 25%, bad weather with 15%, work zones with 10%, and poor traffic signal timing and special events with 5% each one. Since controlling the weather is not a reality and building new road infrastructure is a slow process, modern societies need to rely on new technologies to avoid congestion and its related problems.

One such technology is the Traffic Management System (TMS), which comprises a set of applications and management tools with the aim of improving the overall efficiency of transportation systems by integrating information, communication and sensory technologies [1]. In summary, TMSs collect traffic related data from

* Corresponding author.
E-mail addresses: allanms@lrc.ic.unicamp.br, souza.m.allan@gmail.com (A.M. de Souza), sadao@lrc.ic.unicamp.br (R.S. Yokoyama), boukerch@site.uottawa.com (A. Boukerche), jgmm@dcc.ufmg.br (G. Maia), cerqueira@ufpa.br (E. Cerqueira), loureiro@dcc.ufmg.br (A.A.F. Loureiro), leandro@ic.unicamp.br (L.A. Villas).

http://dx.doi.org/10.1016/j.comnet.2016.09.011
1389-1286/

heterogeneous sources, integrate such data into useful information and use the resulting information to provide applications to drivers with the final goal of detecting, controlling and reducing congestion. One building block for the realization of modern TMSs is the Vehicular Ad hoc Network (VANET), which is composed of mobile nodes (vehicles) with embedded sensors, processing units, and wireless interfaces [5,6]. On such network, vehicles cooperate with one another to create an ad hoc network. Therefore the existence of fixed communication infrastructure is not a requirement. However, fixed Road Side Units (RSU) are often used to improve the network capacity of VANETs. In general, VANETs act as the sensing, communication and actuation platforms of modern TMSs [1].

Many traditional TMSs have been proposed in the literature to detect, control, and reduce congestion in urban scenarios as discussed below. In these solutions, statistics, such as traffic density, speed of vehicles and travel times are collected and sent to a central server in order to detect traffic jams. Thereafter, new routes for the vehicles are calculated and disseminated to drivers as a way to avoid the problematic areas [2,7–15]. However, most of these solutions are able to take action only after a traffic jam is already in place (non pro-active). Moreover, they exchange traffic information inefficiently (disseminate traffic information to unaffected geographical areas). They also assume that vehicles have full 3G or 4G Internet connectivity (e.g., WAZE) to provide their services.

In order to overcome the aforementioned limiting factors of related TMSs solutions, we propose ICARUS (**I**mprovement of traffic **C**ondition through an **A**lerting and **R**e-ro**U**ting **S**ystem), which relies on VANETs to offer a lightweight and distributed solution for controlling and reducing traffic congestion in urban environment, i.e. the vehicle sends its position and retransmits the information only when it receives an alert. After receiving a traffic event from an alert generation system, ICARUS operates under two main phases: *i)* Information Dissemination and *ii) Re-routing. In the Information Dissemination phase, the vehicle disseminates the alert message to all vehicles in the affected geographical area*. Due to frequent changes in the network topology and density caused by the high mobility of vehicles and the short-range communication of VANETs, this phase presents many challenges. For instance, the broadcast storm problem, which takes place whenever multiple vehicles attempt to transmit simultaneously [16], causing high data traffic, network congestion, packet collisions, service disruption and extra delay at the medium access control (MAC) layer. Another challenge is related to the resynchronization effect caused by the multichannel operation of the IEEE 802.11p standard [6,17]. Indeed, all data dissemination protocols for VANETs that assign different waiting delays to rebroadcast (desynchronization) in an attempt to avoid the broadcast storm problem are vulnerable to this resynchronization effect [18,19]. In summary, these challenges limit the use of existing data dissemination protocols in VANETs [20]. Thus, new protocols must be designed by taking into account the inherent characteristics of VANETs. Finally, at the Re-routing phase, ICARUS computes and suggests new routes to the drivers in order to avoid congested areas. As a result, the steps taken by ICARUS bring many benefits for the society and environment, including the reduction of travel time, fuel consumption, and CO emissions.

To evaluate ICARUS, we conducted a series of simulation experiments to highlight its benefits and how it impacts road traffic performance. To find new routes for the drivers, ICARUS was implemented together with three typical shortest path routing algorithms: A*, Dijkstra and Probabilistic k-Shortest Path. ICARUS reduces the travel time in about 68%, and, consequently, the fuel consumption in 48%, and the CO emission in 48%, when compared to scenarios without the ICARUS support.

This work is organized as follows. Section 2 discusses the literature related to congestion minimization in urban environments. Section 3 describes ICARUS. Section 4 presents the performance evaluation results. Finally, Section 5 concludes the study and discusses some future work.

2. Related work

We discuss the related work according to the phases of ICARUS. Therefore, we first describe approaches related to data dissemination, and then we present solutions for congestion detection and re-routing in VANETs. We discuss the relevant algorithms for data dissemination in the Section 2.1 and the strategies to identify congested areas and re-routing in Section 2.2.

2.1. Data dissemination

Flooding is the simplest way to perform data dissemination, however it leads to the well-known broadcast storm problem when the network is dense. Many data dissemination solutions have been proposed in the literature to overcome such problem. These solutions mainly focus on packet forwarding strategies that employ different parameters, such as position, distance, local topology and expected delay. Some of these solutions are described hereafter.

Adaptive Information Dissemination (AID) [21] is a distributed statistically-based broadcast suppression protocol for VANETs. In AID, each vehicle counts the number of redundant messages received from its neighbors. Based on the inter-arrival time between message receptions, a vehicle decides whether to rebroadcast a message or not. For instance, in a high-density traffic scenario, after receiving some redundant retransmissions for a given message, a vehicle may decide not to rebroadcast it, assuming it was already transmitted by many other vehicles. The protocol does not use any neighbor information or any kind of infrastructure. However, it works only on networks with no partitions.

Distance Based Relay Selection (DBRS) [22] is a simple and efficient strategy used to disseminate information in a network. Upon receiving a data packet, the vehicle holds it for a time interval inversely proportional to the distance to the destination vehicle. Thus, it is preferable to use vehicles situated further from the transmitting vehicle to disseminate information. When a vehicle scheduled to retransmit a packet overhears the retransmission of this same packet from another vehicle, it cancels its own retransmission to avoid the broadcast storm problem. This approach is efficient in handling the broadcast storm problem, however it is prone to two other problems. The first one refers to the high delay, since there is no guarantee of the existence of vehicles close to the communication radius (the ones that will transmit with the lowest delay). The second problem refers to the coverage that can be low, since vehicles will cancel their retransmission indiscriminately upon hearing the retransmission of the same packet.

Data Dissemination Protocol in Vehicular Networks (DRIVE) [23] performs data dissemination on both dense and sparse networks. In a dense network, DRIVE selects only the highest priority vehicle within the transmission range of the sender vehicle to continue the dissemination process. The highest priority vehicles are located in a region known as the sweet spot, the same employed in the GEDDAI protocol [24]. In a sparse network, whenever a network partition is detected by a source vehicle, DRIVE employs the recovery zone concept. Vehicles outside the area of interest are used to disseminate data about the event within the area of interest. Those vehicles outside the area of interest (AoI) form a recovery zone. The main purpose of using a recovery zone is to perform data dissemination for vehicles separated by network partitions that are within the AoI.

Similarly to DRIVE, our data dissemination protocol also selects vehicles inside pre-defined high priority zones in order to

Table 1
A qualitative comparison of data disseminations solutions.

Routing protocol	Forwarding strategy					Objectives			Scenario	
	Position based	Statistical based	Store-carry-forward	Distance based	Timer based	Broadcast storm	Network partition	Aware of the channel switching	Highway scenarios	Urban scenarios
AID [21]		√			√	√				√
DBRS [22]	√			√		√				√
DRIVE [23]	√		√	√	√	√	√		√	√
ICARUS	√		√	√	√	√	√	√	√	√

retransmit the messages. However, contrary to DRIVE, our solution is completely aware of the channel switching mechanism employed by the IEEE 802.11p MAC protocol, by monitoring the activated channel. This means that our protocol is not affected by the resynchronization effects caused by the channel switching mechanism, which affects most data dissemination solutions for VANETs, as shown in [6,25].

Table 1 presents a qualitative comparison of data dissemination solutions for VANETs discussed previously, and also includes ICARUS. The classification is based on three sets of criteria: forwarding strategy, objectives and scenarios.

2.2. Congestion detection and re-routing

The related proposals available in the literature focus on identifying congested areas reactively. This typically happens by collecting data from vehicles and roads, processing them to determine the slowest/fastest roads and which ones are congested. With this information, vehicles are rerouted considering the best route for each one of them. Pan et al. [7] propose a centralized system to acquire in real time the vehicle geographic position, speed and direction to detect traffic jam. Once detected, vehicles approaching the traffic jam are re-routed based on three different algorithms. First, Dynamic Shortest Path (DSP) proposes a route to the shortest path with the lowest travel time, but this algorithm has a shortcoming, which is the possibility to move the congestion to another spot. Second, Random k Shortest Paths (RkSP) chooses randomly a route among k shortest path routes. The goal of this algorithm is to avoid switching congestion from one spot to another one by balancing the re-routed traffic among several paths. Third, Entropy Balanced k Shortest Paths (EBkSP) improves RkSP considering the impact that each one of the k routes has on the future of the traffic density. The results show a decrease in the average travel time of 36% to DSP, 41% to RkSP and 45% to EBkSP to the tested scenarios. These schemes do not implement a real-time mechanism to infer when a congestion occurs, only detecting it in the next re-routing interval. Furthermore, these schemes perform the data collection in a centralized way.

Brennad et al. [11] propose a TMS that collects traffic information in real-time and attempts to detect and manage traffic congestion. In such solution, a set of RSUs is distributed through the map in order to provide full coverage of a city. Under this scheme, each RSU is responsible for managing the subset of vehicles and detecting congestion only within its coverage area. Moreover, such proposal includes a congestion control mechanism, which periodically performs the re-routing of all vehicles according to the traffic information collected in a previous step. Similarly to [7], this scheme does not detect congestion as soon as its occurs, since it only detects traffic jams during the next re-routing phase.

Pascale et al. [26] propose a cooperative method for large-scale traffic estimation based on the partition of the road network into smaller networks. In this proposal, each small network estimates its own traffic and then updates it by considering traffic information from neighboring networks based on a stochastic method. This model needs to be calibrated and, eventually, validated on real data and more realistic scenarios to show the benefits provided by the processing distribution to the small networks.

In this study, we propose a solution to avoid the congestion using a distributed application, with the advantage of using an efficient data dissemination mechanism. In our solution, before vehicles relay the alert to neighboring vehicles, each vehicle verifies if its current route will pass through the congestion area. If necessary, takes a new route to prevent that congestion becomes progressively worse.

3. ICARUS

In this section, we describe ICARUS – a TMS that uses data about traffic events to alert vehicles inside an Area of Interest (defined by the application) using our proposed vehicle-to-vehicle data dissemination protocol. The main goal of ICARUS is to redistribute more effectively the road traffic to minimize vehicle congestion in urban centers. In addition, unlike most solutions found in the literature, ICARUS does not require that all vehicles periodically send a message to a central server, which might bring forth very serious impacts on the communication network capacity. To this end, only the set of vehicles inside the Area of Interest (AoI) that will pass through a congested area send messages to a central server.

Definition 3.1 (Problem definition)**.** Considering a VANET environment where the road network is a directed and weighted graph $G = (V, E)$, where the set $V = \{v_1, v_2, \ldots, v_i\}$ corresponds to the set of intersections (vertices), while the set $E = \{e_1, e_2, \ldots, e_i\}$ corresponds to the set of road segments (edges). Moreover, $W = \{w_1, w_2, \ldots, w_i\}$ is a set of weights representing the traffic condition $\forall e | e \in E$. Let $N = \{n_1, n_2, \ldots, n_i\}$ be a set of vehicles (nodes), $R = \{r_1, r_2, \ldots, r_i\}$ a set of routes $\forall n | n \in N$, where $\forall r \subset E^2 | r \in R$. Let $TE = \{t_1, t_2, \ldots, t_i\}$ be a set of traffic events. When a vehicle n_j receives a traffic event t_j, it creates an alert message MSG which is composed by the set of roads that may potentially be affected by the traffic event, thus characterizing a congested area CA. In other words, the affected area is composed by the set of adjacent roads of the traffic event location. Moreover, MSG is used to warn the vehicles inside the AoI, while CA is the set of affected roads that regards a specific traffic event t_j identifying a congested area, CA $\subset E^2$. On the other hand, when a vehicle n_j inside the AoI receives a message MSG, it verifies if $r_j \cap$ CA (i.e., the vehicle verifies if its route will pass through a congested area). Finally, if necessary, an alternative route is computed for the vehicle n_j to avoid the congested area CA.

The ICARUS structure and how each module interacts is presented in Fig. 1. ICARUS receives information about traffic events from other systems, such as congestion detection, accident notification and congestion prediction (e.g., OBD 2 system[1]) Thereafter,

[1] **On Board Diagnostic 2**: provides data of the vehicle in real time (e.g., speed, air bag activated, blinkers on, and engine damage) enabling the application to be aware of the accident.

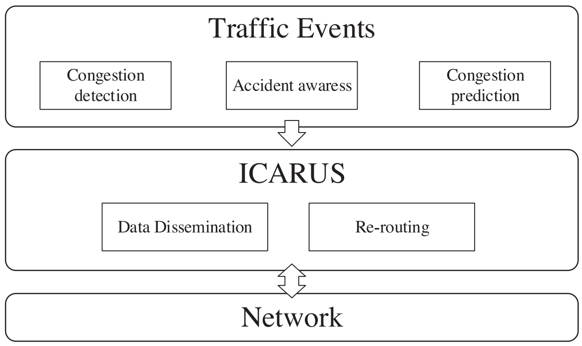


Fig. 1. ICARUS structure.

ICARUS can extract information from traffic events to characterize a congested area, create and disseminate a message to alert all vehicles within an AoI using a V2V data dissemination protocol. Furthermore, the data dissemination needs to address the broadcast storm problem by minimizing the packet collisions during the transmissions, which will also lead to a low overhead, short delays and high coverage. Finally, upon receiving an alert, a vehicle can make a real-time decision by verifying if it will pass through a congested area and, if necessary, the vehicle can change its route using a re-routing algorithm to avoid the congested area. It is worth noticing that ICARUS can be proactive or reactive, depending on the Traffic Event (see Fig. 1). For instance, if the congestion prediction triggers an event with the imminence of congestion, ICARUS acts pro-actively. On the other hand, for an unpredictable event, such as an accident, ICARUS acts reactively.

To address the broadcast storm problem and maximize the coverage, ICARUS uses the concept of a sweet spot.

Definition 3.2 (Sweet Spot)**.** A sweet spot is defined as an area in which its vehicles are best suited to continue performing data dissemination. In other words, among all vehicles that received data to be forwarded, the transmission of a single vehicle within the sweet spot is sufficient to perform the data dissemination efficiently. Vehicles located within the sweet spots are more likely to spread the information further and reach a larger number of neighbors that could not be reached by the previous transmitter [23]. It is worth noticing that, as shown in [23], the use of a sweet spot is also able to handle the network partition problem, however, such problem is out of the scope of this paper.

To minimize the number of collisions by avoiding the synchronization introduced by the IEEE 802.11p standard [6,25], ICARUS implements a desynchronization mechanism that verifies if the computed waiting time to retransmit will lead to a transmission when the control channel is active. If this is the case, ICARUS adds an extra delay to allow the vehicle to transmit in the service channel. The extra delay is at most 50 ms (the time the IEEE 802.11p standard uses to swap from the control channel (CCH) to the service channel (SCH)). In this scenario, if a transmission is scheduled when the control channel is active the extra delay is added to allow the transmission to occur during the SCH.

Furthermore, it is important to stress that ICARUS employs an AoI to each traffic event in order to warn only the vehicles close to the traffic event. In this way, ICARUS prioritizes the vehicles that have higher probability to pass through a traffic jam i.e. the vehicles that should be re-routed. However, the traffic event may affect vehicles farther than the AoI, but is not necessary to alert them at this moment, once that the traffic jam may disappears before the vehicle arrives at the congested road. Otherwise, if the traffic jam still exists, the vehicles will be alerted as soon they enter in the AoI. At last, AoI reduces unnecessary transmissions outside the AoI contributing to reduce the broadcast storm problem.

Fig. 2 describes the steps taken by ICARUS. It starts when a vehicle n_j receives an alert message MSG. Vehicle n_j verifies if it is inside of the AoI (label "A" in Fig. 2). If it is not the case, vehicle n_j verifies if the received MSG is already scheduled to be rebroadcast (label "C" in Fig. 2) to decide if it cancels the scheduled MSG and/or discards the received message MSG. Otherwise, vehicle n_j verifies if it will pass through the *CA* (label "F" in Fig. 2). If it is the case, an alternative route is computed for vehicle n_j to avoid the congested area *CA*. The algorithms (Dijkstra, A* and Probabilistic k-Shortest Path) to calculate an alternative route are explained later. Moreover, vehicle n_j checks whether it is the first time it received the message MSG (label "B" in Fig. 2). If so, vehicle n_j computes the waiting time to rebroadcast and continues the dissemination process based on the *sweet spot* and its position (label "D" in Fig. 2). Moreover, after calculating the waiting time, vehicle n_j verifies if the CCH will be active in the computed waiting time (label "E" in Fig. 2). If it is not the case, the vehicle schedules the retransmission for the computed waiting time. Otherwise, the vehicle adds an extra delay to the computed waiting time to avoid transmitting the message when the control channel is active, i.e., it avoids transmitting the message when the SCH is active. Finally, the vehicle schedules the retransmission for the computed time (waiting time plus an extra delay). Please refer to Algorithm 1 for more details concerning the sweet spot (Lines 3–12), how the waiting time is calculated and how the extra time is added to the calculated waiting time (Lines 12–24).

Algorithm 1: Checks whether the receiving node is within the sweet spot and computes the waiting time to schedule the packet re-transmission and adds an extra time to the *Delay*, if necessary, to not transmit in the control channel.

1 **Input**: (x_s, y_s) and (x_r, y_r) `// Coordinates of the transmitting and receiving nodes`
2 **Output**: *Delay* `// Computed waiting time to schedule the transmission`
`// The two-argument function atan2 is a variation of the arc-tangent function`
3 $angle \leftarrow \text{atan2}(y_s - y_r, x_s - x_r)$;
4 $distToSender \leftarrow \sqrt{(x_s - x_r)^2 + (y_s - y_r)^2}$;
5 $defaultDelay \leftarrow 0.01 \times (communicationRadius - distToSender)$;
6 **if** $((angle \geq 67.5^\circ$ *and* $angle \leq 112.5^\circ) \parallel$ $(angle \geq 157.5^\circ$ *and* $angle \leq 202.5^\circ) \parallel$ $(angle \geq 247.5^\circ$ *and* $angle \leq 292.5^\circ) \parallel$ $(angle \leq 22.5^\circ \parallel angle \geq 337.5^\circ))$ **then**
`// Calculate the waiting time for priority 1`
7 $Delay \leftarrow defaultDelay + \text{random}(0, 0.01)$
8 **end**
9 **else**
`// Calculate the waiting time for priority 2`
10 $Delay \leftarrow defaultDelay + \text{random}\ (0.02, 0.05)$
11 **end**
`// Adds an extra delay to avoid the attempt of transmit in the control channel`
12 **if** *CCH is active* **then**
13 $cch\_cycles = \lfloor \frac{Delay}{0.05} \rfloor$;
14 $extraTime \leftarrow T_s + (cch\_cycles \times 0.05)$;
15 $Delay \leftarrow Delay + extraTime$;
16 **end**
17 **else**
18 $T_{tmp} \leftarrow Delay - remain\tilde{\ }time\tilde{\ }to\tilde{\ }change\tilde{\ }thechannel$;
19 **if** $T_{tmp} > 0$ **then**
20 $chh\_cycles \leftarrow \lceil \frac{T_{tmp}}{0.05} \rceil$;
21 $extraTime \leftarrow cch\_cycles \times 0.05$;
22 $Delay \leftarrow Delay + extraTime$;
23 **end**
24 **end**

To compute an alternative route and re-route the vehicles to avoid a congested area, the re-routing mechanism of ICARUS im-

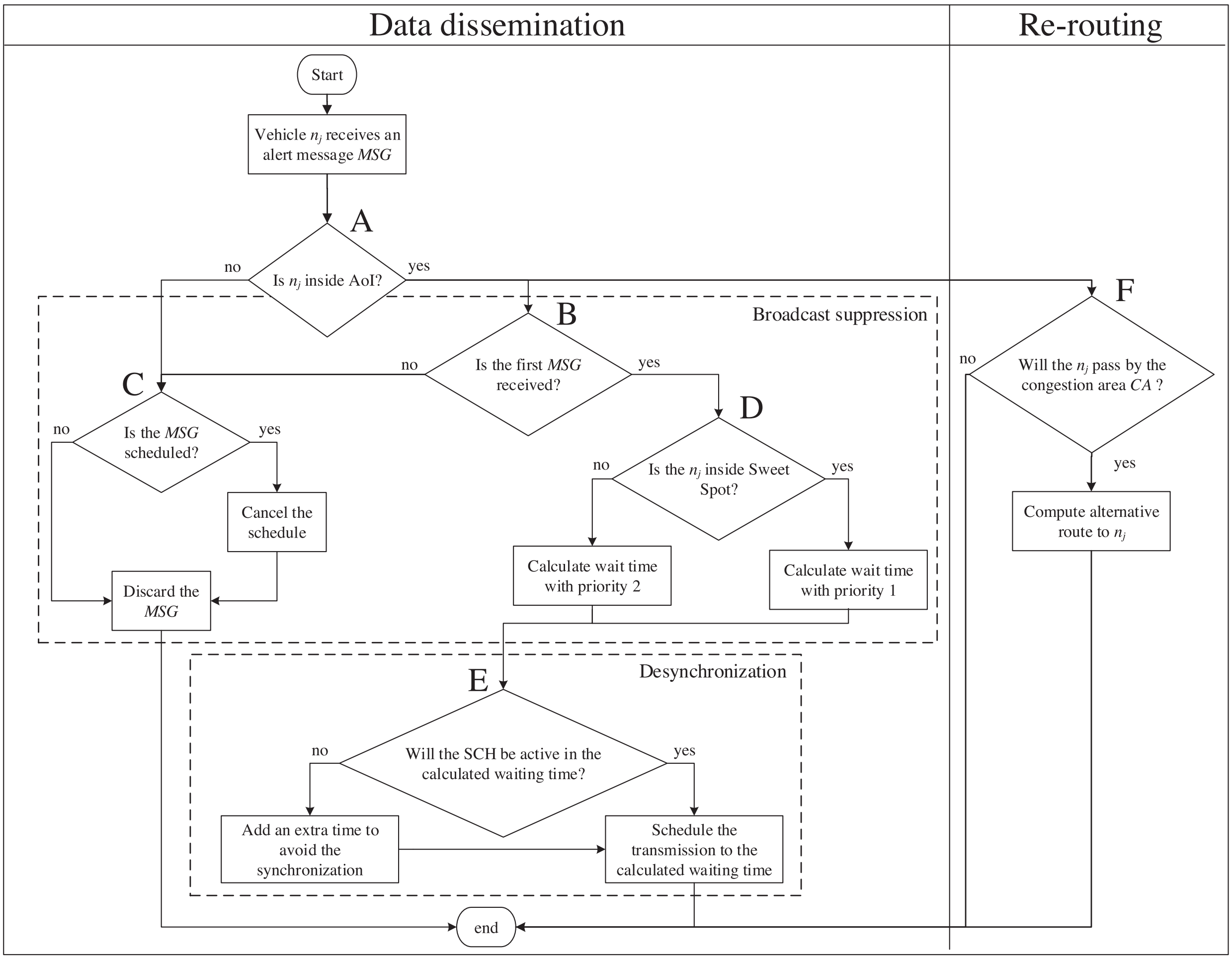


Fig. 2. Procedures of ICARUS.

plements three different routing algorithms (Dijkstra, A* and Probabilistic k-Shortest Path) to avoid the congested areas. These are shortest path algorithms, however each algorithm uses different metrics to select a route.

1. *Dijkstra:* finds the path with the lowest length cost from the current vehicle position to its destination. Dijkstra is one of the optimal algorithms based on the labeling method. However, the algorithm only uses the length of each road as re-routing metric, thus as the traffic becomes denser, Dijkstra algorithm may causes a new traffic jam in a different area (i.e. it can creates a new congested area), because in dense scenarios many vehicles will be routed to the same route.
2. *A*:* is a variant of Dijkstra's algorithm and the its time complexity depends on the heuristic. Like [27], in our implementation, A* uses the traffic condition in each road to calculate the shortest path. Hence, A* restricts the search space and reduces the computational time. In a road traffic application, the search space is restricted to the area where traffic congestion occurs [28]. A* uses the traffic condition between the current location and the destination as a heuristic function. This heuristic function reduces the probability of occurring a new congestion in a different area.
3. *Probabilistic k-Shortest Path (PkSP):* uses the Boltzmann probabilistic algorithm [29] to select the route to the vehicle. This approach attributes a weight to each route of the calculated *k* set calculated. The weight is based on the total route traffic condition and represents the probability to be selected, i.e., the routes with the lowest weight are more likely to be selected. In other words, to choose the route among the *k* shortest paths in order to balance the traffic, the *Probabilistic k-Shortest Path Algorithm* computes the sum of the weights for all *k* routes. The routes with the lowest total weight are the most appropriate and the set R_n is used to decrease the likelihood for them being always chosen, thus balancing the load among the roads. The decision rules to choose the new route are based on the following equations, where N is the set of vehicles on the map; R_n is the set of paths of the vehicle n; r_n^i is the path i of vehicle n and $(r_n^i \in R_n)$; w_n^i is the weight of path r_n^i.

The $Norm(w_n^i)$ is normalized value of w_n^i ($w_n^i \in [0, 1]$) defined by Eq. (1).

$$Norm(w_n^i) = \frac{W(r_n^i)}{max\{W(r_n^i) \quad | \quad \forall r_n^i \in R_n\}}. \tag{1}$$

The K_T^n is the Boltzmann constant of vehicle n for temperature T, according to Eq. (2):

$$K_T^n = \sum_{i \in R_n} e^{-\left(Norm(w_n^i)/T\right)}. \tag{2}$$

$P_T^n(r_n^i)$ is the probability of choosing path i of vehicle n with the parameter of temperature T, according to Eq. (3):

$$P_T^n(r_n^i) = \frac{1}{K_T^n} e^{-\left(Norm(w_n^i)/T\right)}. \tag{3}$$

When $T \to \infty$, all candidate paths have the same probability of being chosen, i.e., the process approaches a uniform random distribution. When $T \to 0$, the path with a lower weight has a higher probability of being chosen.

$E(R_n)$ is the path chosen ($E(R_n \in R_n)$) according to Eq. (4), in addition a random variable X which vary from 0 to 1 was used to provide a better traffic balance.

$$E(R_n) = \max\{X \times P_T^n(r_n^i) | \forall r_n^i \in R_n, X \sim \cup([0,1])\} \quad (4)$$

4. Performance analysis

The assessment of ICARUS is divided into two parts: data dissemination, and congestion avoidance and re-routing. First, the data dissemination evaluation compares the data dissemination mechanism of ICARUS with four solutions presented in the literature: Flooding, AID [21], DBRS [22] and DRIVE [23]. Second, the congestion avoidance and re-routing evaluations are further divided into four distinct evaluations: (i) we evaluate the literature solutions DSP and RkSP in our simulation scenario, in order to identify the best parameter values for these protocols for a later comparison with ICARUS; (ii)we compare ICARUS performance under three different shortest path algorithms: Dijkstra, A* and probabilistic k shortest path; (iii) we compare ICARUS with literature solutions in a congested scenario caused by a high traffic demand, and; (iv) we compare ICARUS with literature solutions in a congested scenario caused by accidents.

4.1. Simulation tools

To conduct the performance analysis, we have implemented ICARUS in the Network Simulator OMNeT++ 4.3 [30]. Moreover, we employ SUMO (*Simulator for Urban MObility*) [31], version 0.17.0, to manage the mobility of vehicles. For the vehicular network, we use the framework Veins 2.1 [32], which implements the IEEE 802.11p standard and the signal attenuation model caused by obstacles. Finally, the EMIT model [33], which is implemented in SUMO, calculates the CO emissions and fuel consumption of vehicles. EMIT is a statistical model simplified from the HBEFA formula [34] to compute instant CO emissions and fuel consumption based on acceleration and vehicle speed.

4.2. Data dissemination evaluation method

A realistic scenario is applied to the simulations, obtained from a real map using the OpenStreetMap tool [35]. The scenario is a 4 km^2 fragment of Manhattan, USA, with several blocks and two-way streets so the vehicles can move in opposite directions. The vehicle density varies from 300 to 700 vehicles/km^2 using three classes of vehicles: cars, buses and trucks, with proportion of 50% to cars, 25% to buses and 25% to trucks. Furthermore, to generate the vehicles' routes, we use a random mobility model, so that for each replication and density, different routes are generated for each vehicle.

Table 2 summarizes the simulation parameters and the associated values used in our assessment. For instance, the transmission power is set to 2.2 mW. With this configuration, the communication range can reach 300 m using the two-ray ground propagation model [36]. Moreover, we set the parameter T of the re-routing algorithm equal to 1. As shown in [11], when T assumes a big value, the vehicles tend to choose the same path during the re-routing phase, since different cars have similar probabilities of choosing the same alternative route. On the other hand, when T assumes small value, the cars tend to choose the path with the lower weights, thus leading to a better load-balancing of the road network. In fact, according to the simulation assessment presented in [11], when T = 1 leads to the best results. Finally, for every analysis, the results represent the mean of 33 replications with a confidence interval of 95%.

Table 2
Simulation parameters.

Parameters	Values
Transmission power	2.2 mW
Wireless range	300 m
Bit rate	18 Mbit/s
Scenario	4 km^2
AoI	1 km
Number of simulations	33
Confidence interval	95%

After the simulation stabilizes, a vehicle in the center of the map of our simulation scenario generates 100 messages of 2048 bytes and starts the data dissemination process at a rate of 500 kbit/s to all vehicles within the AoI, which has 1 km of radius. For this evaluation, the messages correspond to an emergency warning to all drivers being disseminated using a multi-hop communication.

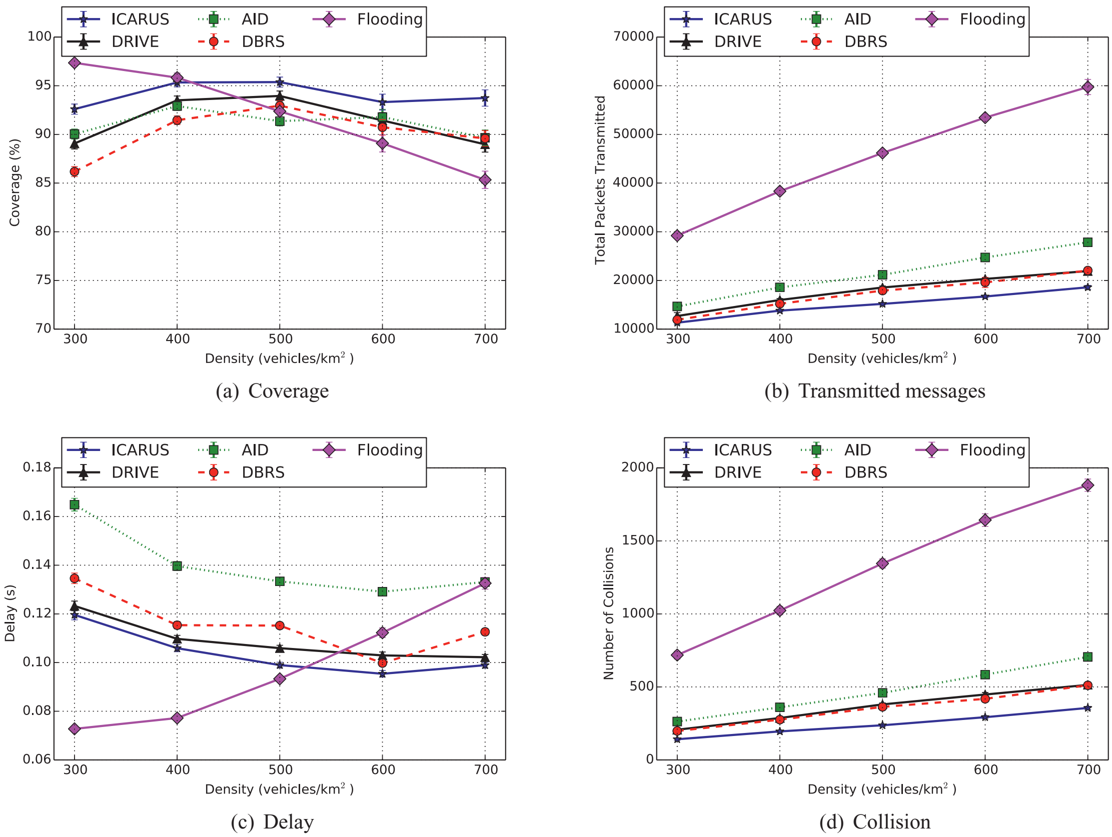
We assess four metrics to evaluate the efficiency, scalability and reliability: (i) *Coverage* is the percentage of vehicles that receive 100% of the data messages being disseminated. It is expected that dissemination protocols achieve a delivery ratio of 100%; (ii) *Transmitted messages* is the total number of data messages transmitted by all vehicles in the network during the dissemination process. A high number of message transmissions is a strong indication that redundant messages are being disseminated, which may result in the broadcast storm problem; (iii) *Delay* is the average time it takes to disseminate the data messages from the source to all vehicles within the AoI. A low delay is of particular interest to time-strict applications, such as warning message dissemination; and, (iv) *Collisions* is the average number of packet collisions at the MAC layer per vehicle to disseminate all data messages. A high number of collisions indicates that a given protocol is not able to avoid the broadcast storm problem.

4.2.1. Data dissemination results

In this section, we discuss the ICARUS data dissemination mechanism. ICARUS was compared with AID [21], DBRS [22], DRIVE [23] and Flooding. We evaluated all protocols under normal and high traffic conditions. The densities used for the simulation were 300, 400, 500, 600 and 700 vehicles/km^2.

Fig. 3 presents all data dissemination results for the simulated scenario. Fig. 3(a) shows the coverage result for all protocols under different traffic densities. For lower densities, Flooding is the only protocol that has a coverage up to 97% in the AoI. This comes from the fact that Flooding essentially rebroadcasts the packet to all vehicles and, in low density, the messages collisions are still a few (see Fig. 3(d)), thus increasing the chance of reaching the best coverage. For the remaining protocols under low densities, only ICARUS has a coverage up to 95%, while the other protocols present a performance from 85% to 93%. However, as the traffic density increases, the data traffic in the network increases as well (see Fig. 3(b)). Such increase in the network traffic leads to more packet collisions, thus leading to a worse coverage for all protocols. On the other hand, ICARUS's efficient broadcast suppression mechanism minimizes the packet collisions, which keeps the protocol's coverage up to 95%.

Fig. 3(b) presents the total number of data packets transmitted. As expected, Flooding is the protocol with the highest overhead. Indeed, in Flooding, all vehicles rebroadcast the message once, thus resulting in redundant retransmissions. ICARUS, by using the proposed broadcast suppression mechanism, disseminates about 68%



(a) Coverage

(b) Transmitted messages

(c) Delay

(d) Collision

Fig. 3. Evaluation of the ICARUS data dissemination protocol.

less packets when compared to Flooding. Moreover, ICARUS disseminates less packets than the other solutions. Therefore, this result shows that ICARUS does not waste bandwidth with unnecessary rebroadcasts.

Fig. 3(c) shows the average delay to disseminate data packets to intended recipients. As can be observed, for lower traffic densities, Flooding has a lower delay, followed by ICARUS. However, as the density increases, ICARUS takes less time to deliver the data packet to all recipients, due to the mechanism to minimize the number of packet collisions by avoiding retransmissions of collided packets. This shows that ICARUS is the best solution, among those evaluated, for applications with strict time requirements, such as alert message dissemination.

Finally, Fig. 3(d) shows the average number of collisions at the MAC layer to disseminate a packet. Flooding has the highest number of packet collisions, which increases as the traffic density increases. This shows that Flooding cannot avoid the broadcast storm problem, especially at higher traffic densities. ICARUS has the lowest packet collisions for all densities, because it implements the desynchronization mechanism to minimize it. Compared to Flooding, ICARUS can reduce the number of average packet collisions by about 80%. Compared to AID, DBRS and DRIVE, ICARUS reduces packet collisions approximately 45%, 30% and 30% respectively.

4.3. Comparison of ICARUS vs. DSP, RkSP, and with routing

In this section, we evaluate the performance of three literature solutions for congestion control: DSP, RkSP and With routing. These solutions require specific parameter setting in order to reach their best performance results. Therefore, in this section, we identify what are the best values for those parameters under our assessment scenario. We investigate the routing interval of DSP, which is described in Section 4.3.1. Moreover, we investigate the parameter k, used to calculate a set of k routes in RkSP, which is described in Section 4.3.2. We also compare ICARUS performance under three different shortest path algorithms: Dijkstra, A* and probabilistic k shortest path, which is described in Section 4.3.3. Finally, ICARUS was compared with OVMT (Original Vehicular Mobility Trace), i.e., the original mobility trace in which no vehicles perform re-routing, DSP [7], RkSP [7] and the solution proposed in [11] (With routing), which is described in Section 4.4.

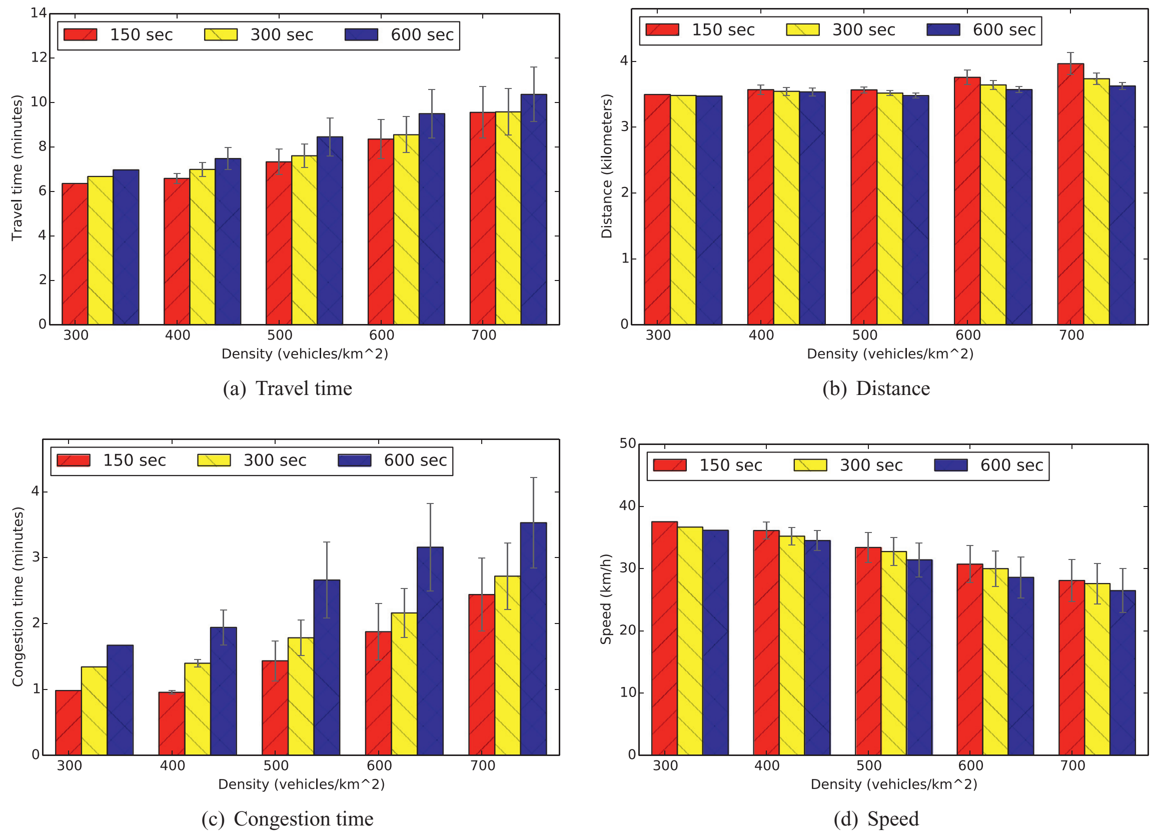
We evaluate all routing algorithms in a Manhattan grid scenario with size of 4 km^2 and we use the same densities as in the data dissemination assessment: 300, 400, 500, 600 and 700 vehicles/km^2. To evaluate the performance of the congestion control solutions, the following metrics are used: (i) *Travel time* is the total time that a vehicle takes to travel its entire route; (ii) *Distance* traveled by the vehicle to go from the origin to its destination; (iii) *Congestion time* is the total time that the vehicle stays stuck in a congestion; and, (iv) *Speed* is the average speed of the vehicle during its trip.

4.3.1. Analysis of routing interval

In this section, we analyze the routing interval parameter. To evaluate it, we used the DSPSP [7] solution, since it does not present any other configuration parameter that may impact its performance. Therefore, the optimal value for the routing interval of DSP may be used with any of the routing solutions available.

To evaluate the routing interval, three different intervals were used: 150, 300 and 600 s. These routing intervals were obtained from [7,11] by taking the values with best performance.

Fig. 4 shows the results for all routing intervals according to all evaluated metrics. In particular, Fig. 4(a) shows the travel time results. As can be seen, in both sparse and dense scenarios, the routing interval of 150 s leads to a lower travel time when compared 300 and 600 s. This happens because DSP only identifies and con-



(a) Travel time (b) Distance

(c) Congestion time (d) Speed

Fig. 4. Evaluation of the routing interval for DSP algorithm.

trols the congestion during the re-routing phase. Therefore, with a routing interval of 150 s, DSP is able to identify and control the congestion earlier than when using a higher routing interval. However, with a routing interval of 150 s, DSP re-routes vehicles more frequently when compared to the other intervals. Consequently, this increases the traveled distance (see Fig. 4(b)).

For instance, in dense scenarios, the traveled distance for the routing interval of 150 s is 8% greater than 300 s, and 11% greater than 600 s. On the other hand, such greater traveled distance contributes to minimize the congestion, as can be seen in Fig. 4(c). For all simulated densities, the routing interval of 150 s has a lower congestion time than the other intervals. In addition, in dense scenarios, the routing interval of 150 s reduces the congestion time in 11% when compared to 300 s and 31% when compared to 600 s. Moreover, vehicles spend only 25% of the travel time in a congestion, while for the other intervals they spend 28% and 33%, respectively. Finally, the efficiency of the routing interval is shown in Fig. 4(d). As can be seen, as the density increases the average speed decreases, due to the fact that the congestion time also increases. However, the routing interval of 150 s has a greater average speed under all densities.

We also evaluated routing interval values smaller than 150 s, however the performance of the solution reduced under all assessed metrics. This occurs because the routes change very frequently and in some cases the vehicles can enter in a loop during the exchange of routes.

4.3.2. Analysis of parameter k

In this section, we analyze the parameter *k*. To evaluate the impact of the route in a set of *k* routes, we relied on the RkSP solution. Therefore, we used the routing interval of 150 s, which leads to the best results for the simulated scenario (see Fig. 4), and three different values for *k*: 3, 5 and 7 routes. These values were obtained from [7,11].

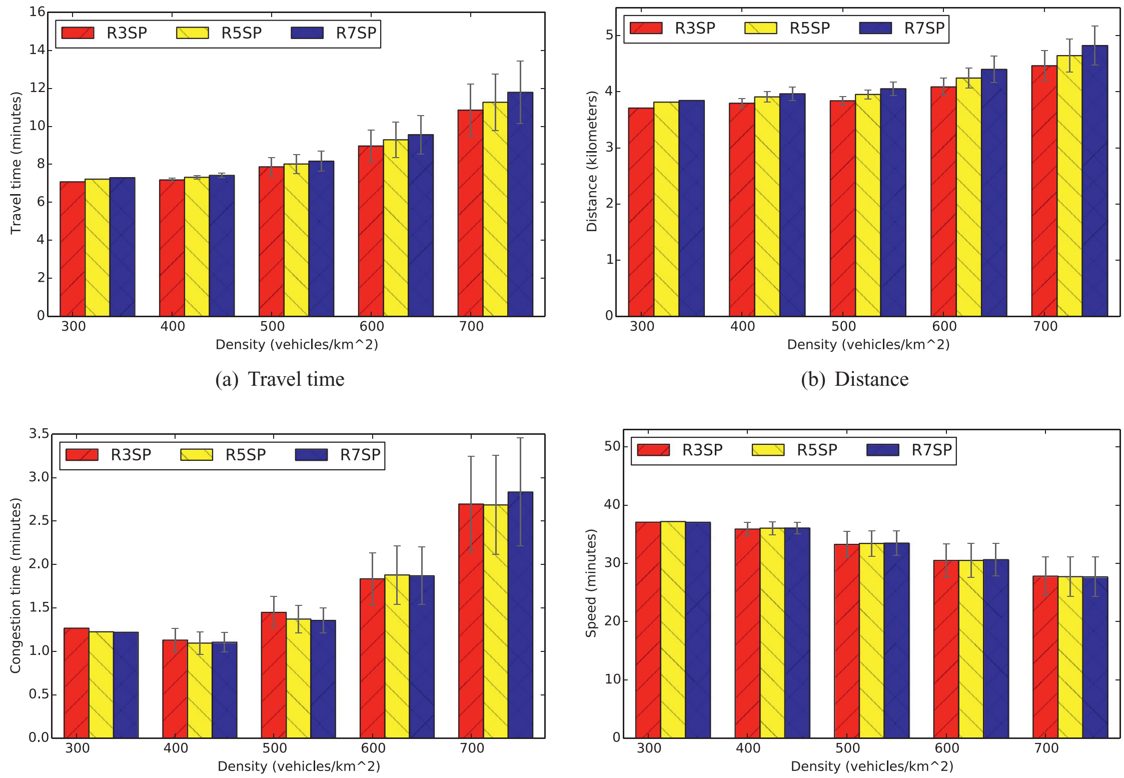
The results are shown in Fig. 5. Fig. 5(a) shows the results for the travel time. Selecting a route from a set of *k* routes increases the travel time and distance, when compared to DSP, which uses the shortest path (see Fig. 4(a) and (b)). However, distributing the routes through a set of *k* possible routes improves the traffic condition. For instance, RkSP reduces the congestion time and increases the average speed (see Figs. 4(c), (d), 5(c) and (d)) when compared to the DSP solution.

Among all values simulated for the parameter *k*, the value 5 leads to the best results. For instance, in dense scenarios, RkSP spends only 19% of its entire travel time in a congestion (see Fig. 5(c)), because with 5 possible routes, the solution can improve the traffic balance. With respect to travel distance, when *k* is 7, it leads to a greater distance when compared to values 3 and 5 (see Fig. 5(b)). Notice, however, that the values 3 and 7 also improve the traffic condition, since they spend approximately 21% of the travel time in a congestion. Moreover, the solution contributes to a smooth traffic flow, since the improvement in traffic condition leads to a better average speed (Fig. 5(d)). All *k* values lead to the same average speed.

We also evaluated values greater than 7 for parameter *k*, however the performance of the solution was worse. This occurs because the probability of choosing the shortest routes reduces.

4.3.3. ICARUS performance evaluation

In this section, we evaluate the performance of our solution. Therefore, to conduct a fair comparison, we evaluate ICARUS in the same grid scenario of DSP and RkSP using the same densities and metrics: (i) Travel time, (ii) Distance, (iii) Congestion time, and (iv) Speed.



(a) Travel time (b) Distance

(c) Congestion time (d) Speed

Fig. 5. Evaluation of the parameter K in RkSP for $k = \{3, 5, 7\}$.

ICARUS needs to receive data about a traffic event in order to characterize a congestion. Therefore, a RSU was positioned at the center of the map. The RSU implements the congestion detection described in [14], which periodically classifies all roads in its coverage. When a congestion is detected, the RSU notifies the vehicles about the traffic event. When the vehicles receive such traffic event, they characterize the congestion area and start the dissemination to notify other vehicles. Finally, when a vehicle receives an alert from other vehicles, it uses the decision making mechanism and, if necessary, changes its route using a re-routing algorithm. Hereafter, we present the results of routing algorithms implemented in ICARUS.

4.3.4. Analysis of routing algorithms implemented in ICARUS

Fig. 6 shows the performance of three routing algorithms implemented in ICARUS: (Dijkstra, A*, PkSP). In particular, Fig. 6(a) shows the travel time results for all routing algorithms. As can be seen, as the density increases the travel time also increases. The Dijkstra algorithm has the greatest travel time for all densities, since it uses the shortest path to recommend new routes. Therefore, as the density increases, most vehicles will use the same shortest paths, creating new congestion points. See Fig. 6(c), where the congestion time for Dijkstra increases faster when compared to other routing algorithms. Differently of Dijkstra, A* uses the traffic condition of the road to calculate the shortest path. As a consequence, A* reduces the travel time in approximately 24% and reduces the congestion time in approximately 64% when compared to Dijkstra.

The routing algorithms P3SP, P5SP and P7SP balance the traffic by selecting a route from a set of k routes, where $k = \{3, 5, 7\}$. This traffic balance contributes to improve the traffic condition. However, it also leads to an increase in the travel distance (see Fig. 6(b)). On the other hand, in P3SP, P5SP and P7SP, vehicles spend only 18%, 20%, and 21%, respectively, of the entire travel time in a congestion. A* also leads to a smooth traffic flow. In A*, vehicles spend only 20% of the travel time in a congestion, while in Dijkstra they spend more than 40%.

Fig. 6 (d) shows the average speed for all routing algorithms. The P3SP and P5SP algorithms have a higher average speed due to the traffic balance implemented in their algorithms. P7SP also implements the traffic balance, however as the number k of alternative routes increases, it computes routes that do not have a good traffic condition.

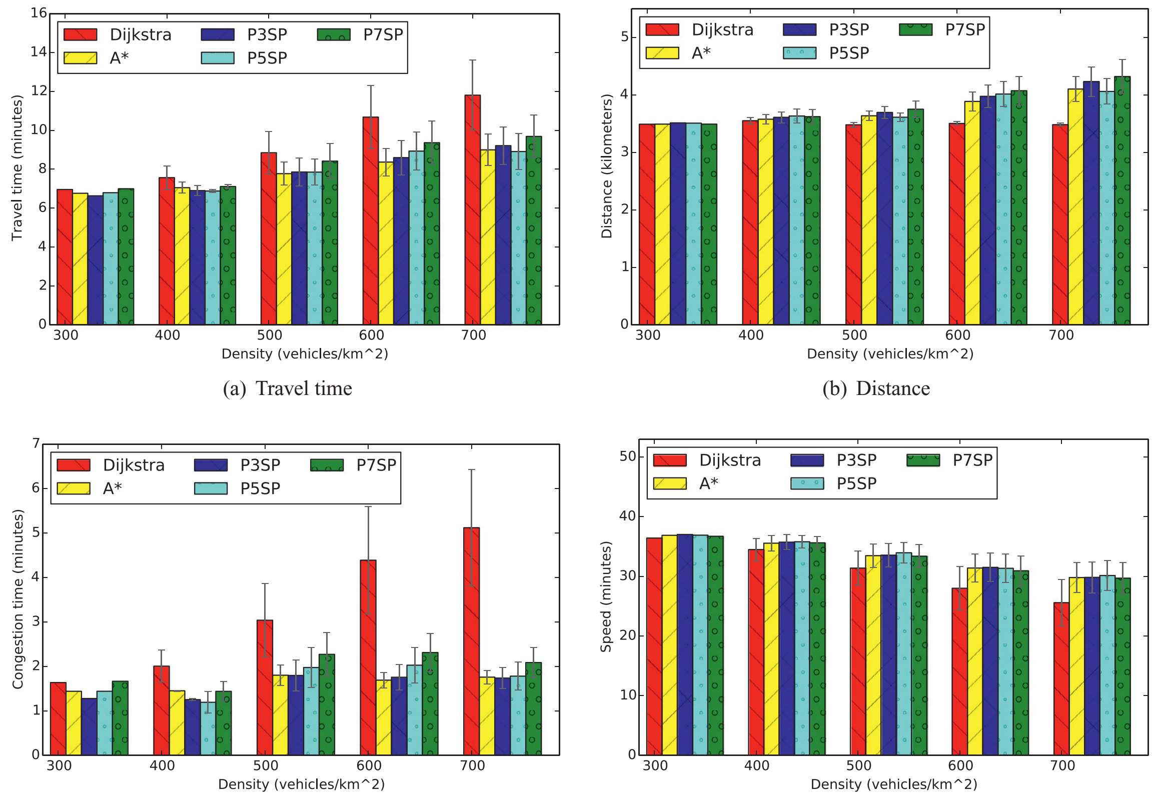
The results shown the efficiency of the re-routing mechanisms implemented in ICARUS. All routing algorithms contribute to a smooth traffic flow. In particular, A* and P3SP reaches the best results. Moreover, P3SP leads to a better traffic condition, because it spends only 18% of the entire trip time in a congestion.

4.4. Performance of ICARUS vs. literature solutions

In this section, ICARUS was compared with OVMT (Original Vehicular Mobility Trace), i.e., real traffic situation, DSP [7], RkSP [7] and the solution proposed in [11] (With routing). Furthermore, two different congested scenarios were simulated. The first one is a congested scenario caused by a high density of vehicles, such as rush hours. Such scenario is described in Section 4.4.1. The second one is a congested scenario caused by a vehicle accident, where the accident blocks the road and creates a congestion. This scenario is described in Section 4.4.2.

4.4.1. Congestion caused by high densities of vehicles

In this section, we evaluate all solutions under a very high traffic condition to force the creation of congestions (density of 1500 vehicles/km^2). Moreover, the parameters routing interval and K were configured with the best values according to the previous analysis (routing interval = 150 s and $k = 5$). For more details, see Section 4.3.



(a) Travel time

(b) Distance

(c) Congestion time

(d) Speed

Fig. 6. Evaluation of the routing algorithms implemented in ICARUS.

Fig. 7 shows the results for all assessed metrics. As can be seen, OVMT has an average traveled distance of 3.52 km (see Fig. 7(c)) and an average traveled time of 15.79 min (see Fig. 7(a)). However, vehicles spend approximately 46% of the traveled time in the traffic jam (see Fig. 7(b)). On the other hand, DSP decreases the average traveled time in approximately 6% when compared to OVMT. This reduction is due to the periodical re-routing of all vehicles. However, this re-routing increases the average traveled distance in 25% (see Fig. 7(c)). Furthermore, DSP re-routes all vehicles using the Dijkstra algorithm. Therefore, many vehicles can be re-routed through the same route, thus creating traffic jams in other areas of the network. As a consequence, vehicles spend 39% of the entire traveled time in a traffic jam (see Fig. 7(b)). Differently from DSP, both RkSP and the With routing distribute the traffic by selecting a route from the set of k possible routes, thus decreasing the probability of creating congestion in another area. However, RkSP increases the traveled time and the traveled distance by 2% and 40%, respectively, when compared to OVMT. Due to the random selection of routes, routes with greater distances can be selected, increasing the traveled time and the traveled distance. Moreover, vehicles spend 36% of the traveled time in the traffic jam.

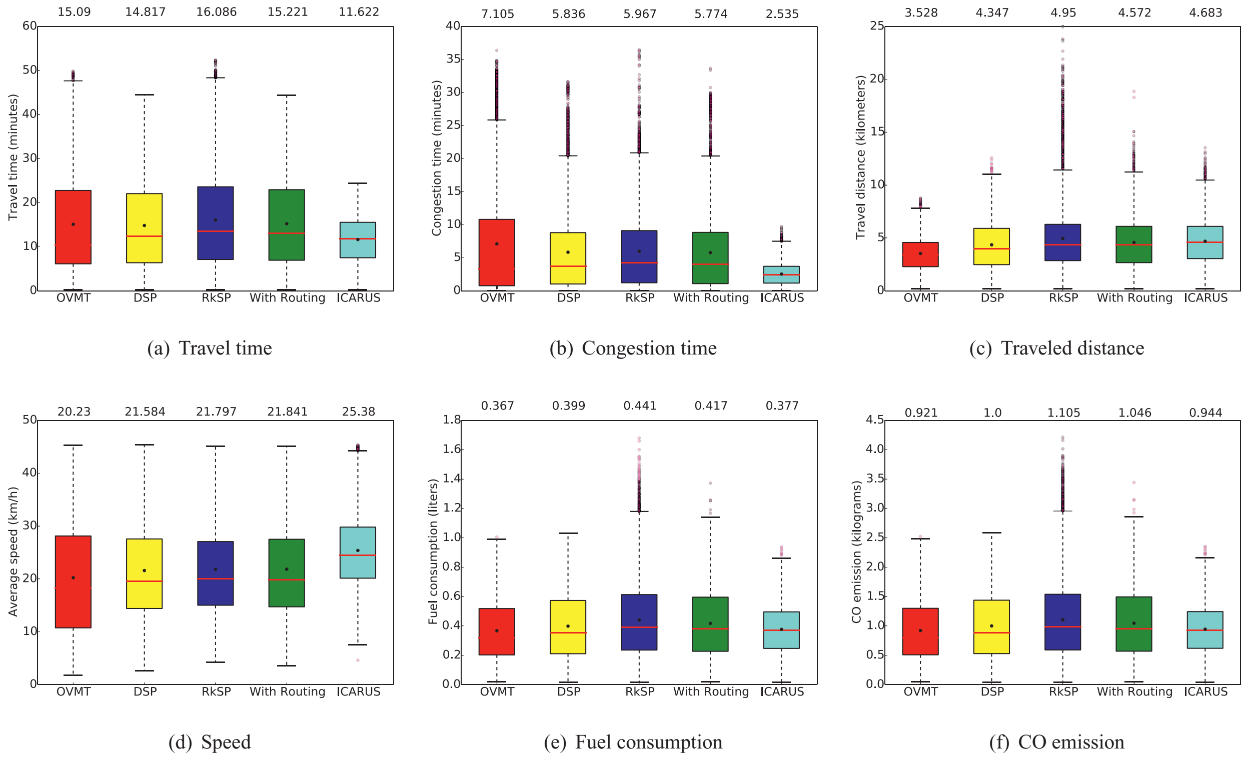
With routing addresses the problem presented in route selection of RkSP by selecting routes with a probabilistic method. Such solution decreases the average traveled time in 3% and increases the average traveled distance in 30% when compared to OVMT. However, it does not totally eliminate traffic jams and vehicles spend 36% of the travel time in a congestion. Finally, ICARUS shows how efficient is to avoid congested areas, because as soon as a congestion is detected, it disseminates the message through an area of interest to inform the approaching vehicles. Furthermore, when a vehicle receives the alert it can use an alternative route to avoid the congested area. In this case, ICARUS decreases the travel time in about 23%, but spends only 20% of the travel time in a traffic jam by increasing the travel distance in 33%. These results show the efficiency of ICARUS when compared to the literature solutions.

The fuel consumption results are directly related to the travel time, congestion time and traveled distance. Fig. 7(e) shows the results for the fuel consumption. As can be seen, OVMT has an average fuel consumption of 0.37 l. DSP, RkSP and the With routing solutions increase the fuel consumption by 9%, 18% and 10%, respectively, when compared to OVMT. Such increase is due to the greater average traveled distance for these solutions. However, ICARUS increases the fuel consumption in only 2% due to a lower traveled distance and a lower time that vehicles spend in the traffic jam. Finally, Fig. 7(f) shows the results for the CO emissions. As expected, the CO emissions are related to the fuel consumption. Therefore, the DSP, RkSP and With routing solutions increase the CO emissions in 9%, 19% and 10%, respectively, when compared to OVMT. On the other hand, ICARUS increases de CO emissions in only 2%.

4.4.2. Congestion caused by accidents

In this section, we evaluated how the solutions behave under traffic jam caused by a car accident. Therefore, an accident was inserted in the scenario in such a way that the accident closes a crossroad blocking four roads in both directions preventing the traffic flow in these roads. Furthermore, the accident has a duration of two hours. The duration of the accident represents the time until the vehicles involved in the accident are removed and the traffic flow in the area is reopened.

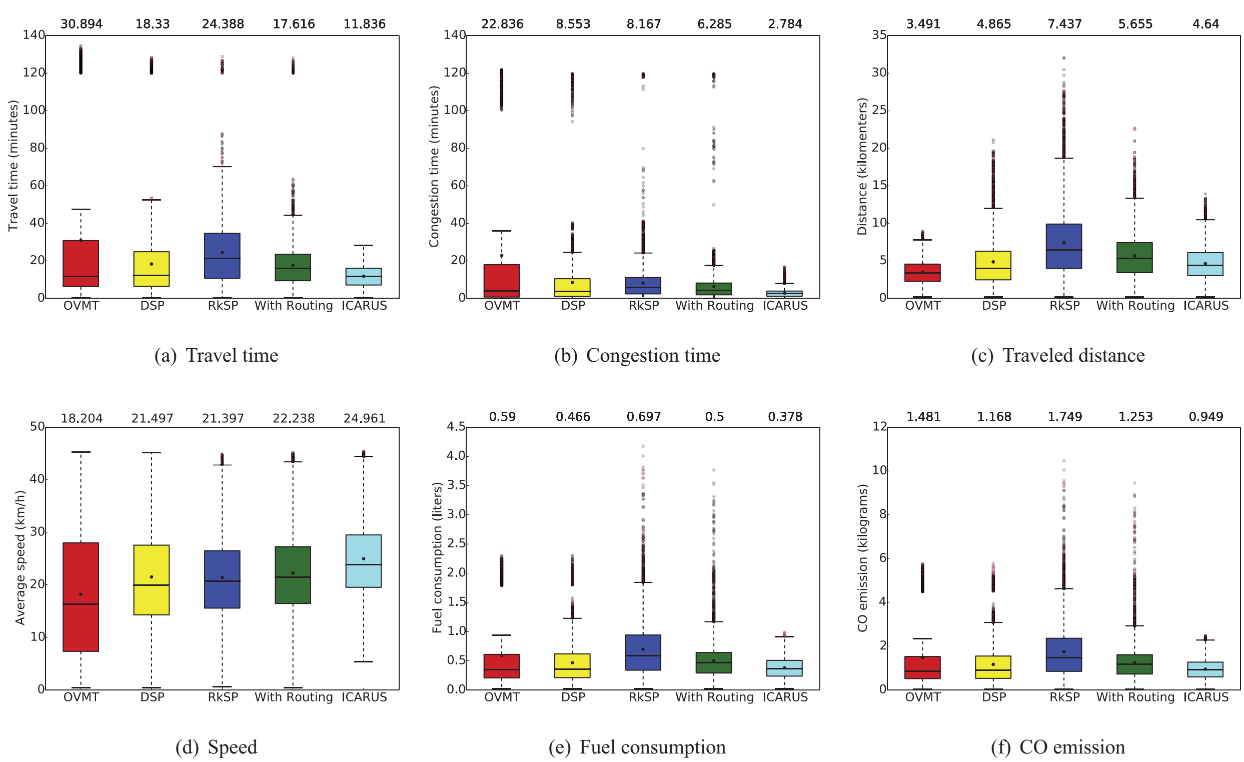
In this section, we evaluated all solutions under a density of 1500 vehicles/km^2). Moreover, the parameters routing interval and K were configured with the best values obtained in the previous analysis (routing interval = 150 s and $k = 5$). For more details, see Section 4.3.



(a) Travel time (b) Congestion time (c) Traveled distance

(d) Speed (e) Fuel consumption (f) CO emission

Fig. 7. Evaluation of ICARUS vs. literature solutions during rush hours.



(a) Travel time (b) Congestion time (c) Traveled distance

(d) Speed (e) Fuel consumption (f) CO emission

Fig. 8. Evaluation of the ICARUS vs. literature solutions in the presence of a car accident.

Fig. 8 shows the results for all assessed metrics. In particular, Fig. 8(a) shows the results for the travel time. As expected, OVMT presents the worst travel time, because it does not have any mechanism to control a traffic jam. Therefore, as the accident blocks the traffic in a crossroad for two hours, many vehicles get stuck in this traffic jam. As a consequence, vehicles have an average congestion time of 22.83 min, that is approximately 74% of the entire average travel time. The routing solutions DSP, RkSP and With routing, have no information about the traffic accident. Therefore, the traffic jam will be detected in the routing phase. However, with no information about the accident many vehicles can enter in the accident road and get stuck in the traffic jam as well (see Fig. 8(a) and (b)).

The solutions are still able to reduce the travel time and the congestion time when compared to OVMT. The DSP solution reduces the travel time in 40%, while the RkSP and With routing reduce the travel time in 21% and 43%. In addition, the congestion time is reduced by 62% for DSP, 64% for RkSP and 72% for With routing when compared to OVMT. Vehicles still spend 46%, 33% and 35% of the entire travel time in a traffic jam for DSP, RkSP and With routing, respectively. On the other hand, the routing solutions increase the traveled distance by 39% for DSP, 112% for RkSP and 61% for With routing (see Fig. 8(c)). The greatest traveled distance by RkSP is due to random route selection, where many vehicles can select long routes to avoid the traffic jam. This is different from them With routing solution, which uses a probabilistic selection to avoid this issue.

ICARUS receives information about traffic events, i.e., it knows about the traffic accident receiving information about it through [15]. Therefore, ICARUS starts controlling the imminence of a traffic jam as soon as it occurs. Accordingly, the vehicles do not enter in roads blocked by the accident, so the vehicles do not get stuck in the traffic jam (see the outliers in Fig. 8(a) and (b)). Furthermore, ICARUS reduces the travel time in 62% when compared to OVMT and 35%, 51% and 32% when compared to DSP, RkSP and With routing. Moreover, ICARUS reduces the congestion time in 88% when compared to OVMT and the vehicles spends only 22% of the entire travel time in the congestion. On the other hand, as in the routing solutions, ICARUS also increases the traveled distance. However, it has the lowest increase, about 30%. These results show that ICARUS leads to a better improvement in the traffic condition when compared to the literature solutions.

The results for the fuel consumption results are directly related to the travel time, congestion time and traveled distance. Fig. 8(e) shows the fuel consumption results. As can be seen, OVMT has an average fuel consumption of 0.59 l. DSP and With routing decrease the fuel consumption in 22% and 15%, while RkSP increases the fuel consumption by 18%. Such increase of RkSP is due to the greater average traveled distance. ICARUS decreases the fuel consumption as well. It presents a reduction of 36% when compared to OVMT. Such greater reduction is because of the lower traveled distance and lower time that vehicles spend in the traffic jam. Fig. 8(f) shows the results for the CO emissions. As expected, the CO emissions are related to the fuel consumption. Therefore, the DSP and With routing decreases the fuel consumption in 21% and 14% when compared to OVMT, while RkSP increases the CO emissions in 18%. Finally, ICARUS decreases the CO emissions in 36%.

The improvement of the traffic condition can be seen in Fig. 8(d), where ICARUS has the greatest average speed. Such result is due to the fast alert dissemination to vehicles and the efficient congestion control mechanism. ICARUS increases the average speed in 37%, 16%, 16% and 12% when compared to OVMT, DSP, RkSP and With routing, respectively.

5. Conclusion

In this work, we proposed a novel Traffic Management System, ICARUS, to minimize the congestion of vehicles in urban centers using a vehicular network. The proposed solution aims to reduce the travel time, congestion time, fuel consumption, CO emissions and maximize the average speed of the vehicles during its trip. Simulation results show that the proposed solution reduces significantly the travel time, fuel consumption and CO emissions. Since, for the travel time, a reduction of approximately 68% was presented, fuel consumption showed savings of 48%, and, finally, CO emissions were reduced by 48%.

Acknowledgments

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

References

[1] S. Djahel, R. Doolan, G.-M. Muntean, J. Murphy, A communications-oriented perspective on traffic management systems for smart cities: challenges and innovative approaches, IEEE Commun. Surv. Tut. 17 (1) (2015) 125–151.

[2] R. Bauza, J. Gozalvez, J. Sanchez-Soriano, Road traffic congestion detection through cooperative vehicle-to-vehicle communications, in: Local Computer Networks (LCN), 2010 IEEE 35th Conference on, 2010, pp. 606–612, doi:10.1109/LCN.2010.5735780.

[3] G. Karagiannis, O. Altintas, E. Ekici, G. Heijenk, B. Jarupan, K. Lin, T. Weil, Vehicular networking: a survey and tutorial on requirements, architectures, challenges, standards and solutions, IEEE Commun. Surv. Tut. 13 (4) (2011) 584–616, doi:10.1109/SURV.2011.061411.00019.

[4] US-DoT, Traffic congestion and reliability: trends and advanced strategies for congestion mitigation, 2015, http://www.ops.fhwa.dot.gov/congestion_report/chapter2.htm.

[5] L.A. Villas, H.S. Ramos, A. Boukerche, D.L. Guidoni, R.B. Araujo, 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, ACM, New York, NY, USA, 2012, pp. 39–46, doi:10.1145/2387027.2387035.

[6] A.M. Souza, G. Maia, 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, 2014, pp. 17–23, doi:10.1109/NCA.2014.7.

[7] J. Pan, M. Khan, I.S. Popa, K. Zeitouni, C. Borcea, Proactive vehicle re-routing strategies for congestion avoidance, in: Distributed Computing in Sensor Systems (DCOSS), 2012 IEEE 8th International Conference on, 2012, pp. 265–272, doi:10.1109/DCOSS.2012.29.

[8] A.M.d. Souza, A. Boukerche, G. Maia, R.I. Meneguette, A.A. Loureiro, 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, MobiWac '14, ACM, New York, NY, USA, 2014, pp. 91–98, doi:10.1145/2642668.2642677.

[9] F. Li, Y. Wang, Routing in vehicular ad hoc networks: a survey, IEEE Veh. Technol. Mag. 2 (2) (2007) 12–22, doi:10.1109/MVT.2007.912927.

[10] S. Dornbush, A. Joshi, Streetsmart traffic: discovering and disseminating automobile congestion using vanet's, in: Vehicular Technology Conference, 2007. VTC2007-Spring. IEEE 65th, 2007, pp. 11–15, doi:10.1109/VETECS.2007.15.

[11] C.A.R.L. Brennand, A.M. de Souza, G. Maia, A. Boukerche, H. Ramos, A.A.F. Loureiro, L.A. Villas, An intelligent transportation system for detection and control of congested roads in urban centers, 2015 IEEE Symposium on Computers and Communication (ISCC) (2015) 663–668. doi: 10.1109/ISCC.2015.7405590.

[12] A.M. de Souza, R.S. Yokoyama, G. Maia, A.A.F. Loureiro, L.A. Villas, Minimizing traffic jams in urban centers using vehicular ad hoc networks, in: 2015 7th International Conference on New Technologies, Mobility and Security (NTMS), 2015, pp. 1–5, doi:10.1109/NTMS.2015.7266505.

[13] A.M. de Souza, R.S. Yokoyama, G. Maia, A. Loureiro, L. Villas, Real-time path planning to prevent traffic jam through an intelligent transportation system, in: 2016 IEEE Symposium on Computers and Communication (ISCC), 2016, pp. 726–731, doi:10.1109/ISCC.2016.7543822.

[14] A.M. de Souza, R.S. Yokoyama, L.C. Botega, R.I. Meneguette, L.A. Villas, Scorpion: a solution using cooperative rerouting to prevent congestion and improve traffic condition, in: Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing (CIT/IUCC/DASC/PICOM), 2015 IEEE International Conference on, 2015a, pp. 497–503, doi:10.1109/CIT/IUCC/DASC/PICOM.2015.71.

[15] A.M. de Souza, R.S. Yokoyama, N.L.S.d. Fonseca, R.I. Meneguette, L.A. Villas, Garuda: a new geographical accident aware solution to reduce urban congestion, in: Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing (CIT/IUCC/DASC/PICOM), 2015 IEEE International Conference on, 2015b, pp. 596–602, doi:10.1109/CIT/IUCC/DASC/PICOM.2015.85.

[16] S.-Y. Ni, Y.-C. Tseng, Y.-S. Chen, J.-P. Sheu, The broadcast storm problem in a mobile ad hoc network, in: Proceedings of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking, MobiCom '99, ACM, 1999, pp. 151–162.

[17] IEEE, Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 6: Wireless Access in Vehicular Environments, IEEE Standards, 2010.

[18] Q. Chen, D. Jiang, L. Delgrossi, IEEE 1609.4 DSRC multi-channel operations and its implications on vehicle safety communications, in: IEEE Vehicular Networking Conference (VNC '09), 2009, pp. 1–8.

[19] M. van Eenennaam, A. van de Venis, G. Karagiannis, Impact of IEEE 1609.4 channel switching on the IEEE 802.11p beaconing performance, in: IFIP Wireless Days (WD '12), 2012, pp. 1–8.

[20] F.D.D. Cunha, A. Boukerche, L. Villas, A.C. Viana, A.A.F. Loureiro, Data Communication in Vanets: A Survey, Challenges and Applications, Research Report RR-8498, INRIA Saclay ; INRIA, 2014. https://hal.inria.fr/hal-00981126
[21] M. Bakhouya, J. Gaber, P. Lorenz, An adaptive approach for information dissemination in vehicular ad hoc networks, J. Netw. Comput. Appl. 34 (6) (2011) 1971–1978, doi:10.1016/j.jnca.2011.06.010.
[22] T.-H. Kim, W.-K. Hong, H.-C. Kim, Y.-D. Lee, Information Networking. Towards Ubiquitous Networking and Services, Springer-Verlag, Berlin, Heidelberg, 2008, pp. 295–304, doi:10.1007/978-3-540-89524-4_30. Chapter An Effective Data Dissemination in Vehicular Ad-Hoc Network
[23] L.A. Villas, A. Boukerche, G. Maia, R.W. Pazzi, A.A. Loureiro, Drive: an efficient and robust data dissemination protocol for highway and urban vehicular ad hoc networks, Comput. Netw. 75 (Part A (0)) (2014) 381–394. http://dx.doi.org/10.1016/j.comnet.2014.10.012. http://www.sciencedirect.com/science/article/pii/S1389128614003673
[24] L. Villas, A. Boukerche, R. Araujo, A. Loureiro, J. Ueyama, Network partition-aware geographical data dissemination, in: Communications (ICC), 2013 IEEE International Conference on, 2013, pp. 1439–1443, doi:10.1109/ICC.2013.6654713.
[25] E. Donato, G. Maia, J.M. Duarte, A.A. Loureiro, E. Madeira, L. Villas, Presync: a method for preventing resynchronization in the ieee 802.11p standard, Computers and Communications (ISCC), 2015 IEEE Symposium on, 2015.
[26] A. Pascale, M. Nicoli, U. Spagnolini, Cooperative bayesian estimation of vehicular traffic in large-scale networks, IEEE Trans. Intell. Transp. Syst. 15 (5) (2014) 2074–2084.
[27] I.-C. Chang, H.-T. Tai, F.-H. Yeh, D.-L. Hsieh, S.-H. Chang, A vanet-based a-star route planning algorithm for travelling time- and energy-efficient GPS navigation App, Int. J. Distrib. Sensor Networks 9 (7) (2013) 1–14.
[28] V.T.N. Nha, S. Djahel, J. Murphy, A comparative study of vehicles' routing algorithms for route planning in smart cities, in: Vehicular Traffic Management for Smart Cities (VTM), 2012 First International Workshop on, 2012, pp. 1–6, doi:10.1109/VTM.2012.6398701.
[29] S. Kirkpatrick, C.D. Gelatt, M.P. Vecchi, et al., Optimization by simmulated annealing, Science 220 (4598) (1983) 671–680.
[30] A. Varga, R. Hornig, An overview of the OMNet++ simulation environment, in: International Conference on Simulation Tools and Techniques for Communications, Networks and Systems & Workshops (Simutools '08), 2008, pp. 1–10.
[31] M. Behrisch, L. Bieker, J. Erdmann, D. Krajzewicz, SUMO - simulation of urban MObility: an overview, in: International Conference on Advances in System Simulation (SIMUL '11), 2011, pp. 63–68.
[32] C. Sommer, R. German, F. Dressler, Bidirectionally coupled network and road traffic simulation for improved IVC analysis, in: Mobile Computing, IEEE Transactions on, vol. 10, 2011, pp. 3–15, doi:10.1109/TMC.2010.133.
[33] A. Cappiello, I. Chabini, E. Nam, A. Lue, M.A. Zeid, A statistical model of vehicle emissions and fuel consumption, in: Intelligent Transportation Systems, 2002. Proceedings. The IEEE 5th International Conference on, 2002, pp. 801–809, doi:10.1109/ITSC.2002.1041322.
[34] M. Keller, P. Wthrich, Handbook emission factors for road transport, 2014, http://www.hbefa.net/e/index.html.
[35] M.M. Haklay, P. Weber, Openstreetmap: user-generated street maps, IEEE Pervasive Comput. 7 (4) (2008) 12–18, doi:10.1109/MPRV.2008.80.
[36] C. Sommer, S. Joerer, F. Dressler, On the applicability of two-ray path loss models for vehicular network simulation, in: Vehicular Networking Conference (VNC), 2012 IEEE, 2012, pp. 64–69, doi:10.1109/VNC.2012.6407446.

Allan M. de Souza received the graduation degree from University Eurípdes de Marilia (UNIVEM), Brazil, in 2013. His received the Master degree from University of Campinas (UNICAMP), Brazil, in 2016, where his currently a Ph.D. student. His research interests are in the field of data dissemination, congestion detection, congestion control and re-routing in VANETs.

Roberto Sadao received the graduation degree from the Universidade de São Paulo (USP), Brazil, in 2009, where his currently a Ph.D. student. He has been a visiting PhD student at Politecnico di Torino, Italy. His research interests are in the field of mobile computing, context-aware systems and IP connectivity management. He is a student member of the IEEE.

Azzedine Boukerche is a full Professor and holds a Canada Research Chair position at the University of Ottawa (uOttawa), Ottawa, ON, Canada. He is a Fellow of the Canadian Academy of Engineering and the Founding Director of the PARADISE Research Laboratory, School of Information Technology and Engineering (SITE), uOttawa. Prior to this, he held a faculty position at the University of North Texas, Denton, TX, and he was a Senior Scientist at the Simulation Sciences Division, Metron Corporation, San Diego, CA. He was also employed as a Faculty Member in the School of Computer Science, McGill University, Montreal, QC, Canada, and taught at the Polytechnic of Montreal, Montreal, QC, Canada. He spent a year at the JPL/NASA, California Institute of Technology, Pasadena, CA, where he contributed to a project centered about the specification and verification of the software used to control interplanetary spacecraft operated by JPL/NASA Laboratory. He has published several research papers in these areas. His current research interests include wireless ad hoc and sensor networks, wireless networks, mobile and pervasive computing, wireless multimedia, QoS service provisioning, performance evaluation and modeling of large-scale distributed systems, distributed computing, large-scale distributed interactive simulation, and parallel discrete-event simulation. He served as a Guest Editor for the Journal of Parallel and Distributed Computing (special issue for routing for Mobile Ad Hoc, special issue for Wireless Communication and Mobile Computing, and special issue for Mobile Ad Hoc Networking and Computing), ACM/Kluwer Wireless Networks, ACM/Kluwer Mobile Networks Applications, and the Journal of Wireless Communication and Mobile Computing. He serves as an Associate Editor of IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, Elsevier Ad Hoc Networks, Wiley International Journal of Wireless Communication and Mobile Computing, Wiley's Security and Communication Network Journal, Elsevier Pervasive and Mobile Computing Journal, IEEE WIRELESS COMMUNICATION MAGAZINE, Elsevier's Journal of Parallel and Distributed Computing, and SCS Transactions on Simulation. He is the Cofounder of the Qshine International Conference on Quality of Service for Wireless/Wired Heterogeneous Networks (QShine 2004). He served as the General Chair for the Eighth ACM/IEEE Symposium on Modeling, Analysis, and Simulation of Wireless and Mobile Systems, and the Ninth ACM/IEEE Symposium on Distributed Simulation and Real-Time Application (DS-RT), the Program Chair for the ACM Workshop on QoS and Security for Wireless and Mobile Networks, ACM/IFIPS Europar 2002 Conference, IEEE/SCS Annual Simulation Symposium (ANNS 2002), ACM WWW 2002, IEEE MWCN 2002, IEEE/ACM MASCOTS 2002, IEEE Wireless Local Networks WLN 03–04; IEEE WMAN 04–05, and ACM MSWiM 98–99, and a TPC member of numerous IEEE- and ACM-sponsored conferences. He served as the Vice General Chair for the Third IEEE Distributed Computing for Sensor Networks (DCOSS) Conference in 2007, as the Program Cochair for GLOBECOM 2007–2008 Symposium on Wireless Ad Hoc and Sensor Networks, and for the 14th IEEE ISCC 2009 Symposium on Computer and Communication Symposium, and as the Finance Chair for ACM Multimedia 2008. He also serves as a Steering Committee Chair for the ACM Modeling, Analysis, and Simulation for Wireless and Mobile Systems Conference, the ACM Symposium on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks, and IEEE/ACM DS-RT. He was the Recipient of the Best Research Paper Award at IEEE/ACM PADS 1997, ACM MobiWac 2006, ICC 2008, ICC 2009, and IWCMC 2009, and the Recipient of the Third National Award for Telecommunication Software in 1999 for his work on a distributed security systems on mobile phone operations. He has been nominated for the Best Paper Award at the IEEE/ACM PADS 1999 and ACM MSWiM 2001. He is a Recipient of an Ontario Early Research Excellence Award (previously known as Premier of Ontario Research Excellence Award), Ontario Distinguished Researcher Award, and Glinski Research Excellence Award.

Guilherme Maia is a Postdoctoral Researcher at the Federal University of Minas Gerais, Brazil. He got his Ph.D. in Computer Science at this same university in 2013. His research interests include distributed algorithms, mobile computing, wireless sensor networks and vehicular ad hoc networks.

Eduardo Cerqueira received his Ph.D. in Informatics Engineering from the University of Coimbra, Portugal, in 2008. He is an associate professor at the Faculty of Computer Engineering and Telecommunications of the UFPA in Brazil. His publications include four books, four patents and over than 120 papers in national/international refereed journals/conferences. He is involved in the organization of several international conferences and workshops, including Future Multimedia Networking, Latin America Conference on Communications, and Latin American Conference on Networking. He has been serving as a Guest Editor for many special issues of various peer-reviewed scholarly journals. His research involves multimedia, Future network, Quality of Experience, Mobility.

Antonio A.F. Loureiro received his B.Sc. and M.Sc. degrees in computer science from the Federal University of Minas Gerais (UFMG), Brazil, and the Ph.D. degree in computer science from the University of British Columbia, Canada. Currently, he is a full professor of computer science at UFMG, where he leads the research group in wireless sensor networks. His main research areas are wireless sensor networks, mobile computing, and distributed algorithms. In the last 10 years he has published regularly in international conferences and journals related to those areas, and also presented tutorials at international conferences.

Leandro A. Villas is an Associate Professor of Computer Science at University of Campinas (UNICAMP), Brazil. He received his Ph.D. in Computer Science from the Federal University of Minas Gerais, Brazil with a partial doctoral fellowship at University of Ottawa, Canada, in 2012. His research interests include localization and synchronization algorithms, distributed algorithms, and wireless ad hoc, vehicular, and sensor networks. He is an author of several papers in the different areas of his research interests.