A Fully-distributed Traffic Management System to Improve the Overall Traffic Efficiency

Allan M. de Souza and Leandro A. Villas

University of Campinas, Campinas, SP, Brazil

allanms@lrc.ic.unicamp.br, leandro@ic.unicamp.br

ABSTRACT

In recent years, the number of vehicles has increased faster than the available infrastructure. Consequently, traffic congestion has become a daily problem affecting several aspects of modern society, including regional economic development. In this way, Traffic Management System (TMS) have been proposed to improve the traffic efficient and minimize traffic congestion problems. These systems rely on gather traffic-related data in a central entity to identify congestion and suggest alternative routes. However such approach adds load in communication channel depending on the traffic density. In this way, this paper introduces FASTER, a fully-distributed TMS to improve the overall vehicle traffic efficiency that does not overloads the communication channel, providing a suitable distributed solution. Simulation results indicate that our FASTER outperforms the assessed solutions in different scenarios and in different key requirements of TMS.

Keywords

Traffic Management System; VANETs; Traffic efficiency

1. INTRODUCTION

Mobility is one of the most challenging concern in large cities around the world. This takes place due to the fast population growth [9]. On the other hand, the transport infrastructure has not grown at the same pace as the number of vehicles. As a consequence of this, traffic congestion has become a daily problem, creating several negative issues for the society, including traffic congestiton, high number of traffic accidents, effects in the economic development, monetary losses and impacts in the environment [12].

To cope with such problem, modern societies rely on Traffic Management Systems (TMS), which are composed of a set of applications and management tools to improve the efficiency and the safety of the transportation system. TMS integrate information, communication and sensory technologies to collect traffc-related data from heterogeneous sources such as vehicles, traffic ligths, in-road and road side sensors and so on [9, 6]. Upon collecting and aggregating such data, they are exploited to identify traffic hazards which may degrade the traffic efficiency, to them deliver services to overcome these hazards.

One building block to develop an efficient TMS is the Vehicular Ad hoc Network (VANET). It considers vehicles as mobile nodes with embedded sensors, processing units, and wireless communication interfaces [12, 19, 18]. Thanks to these features, vehicles can cooperate among themselves and create an ad hoc network to provide and receive data to other entities. In addition, though an infrastructure is not a requirement, it can be used such as Road Side Units (RSU) to improve the its capacity and provide another features. In general, VANETs acts as the sensing, communication and actuation platforms for TMS [9].

Several TMS to improve the traffic efficiency have been proposed [10, 1, 15, 6, 5, 16, 8, 7]. However, these solutions perform the exchange of traffic information in an inefficient way, which may overload the communication network. In general, they also assume that the entire city is covered by RSU in order to detect, control and reduce traffic congestion. However most cities are not fully covered by RSU. Moreover, as far as we are aware, there is no attempt to solve the vehicular congestion problem in a fully-distributed way as does our proposed.

In order to overcome such problem, we introduce the FASTER, a fully-distributed VANET-based system designed to improve the vehicle traffic efficiency. FASTER relies to four main phases: *(i)* Information gathering; *(ii)* Information aggregation; *(iii)* Information dissemination; and *(iv)* Congestion detection and avoidance. It addresses network challenges during the data dissemination process, which includes the *broadcast storm problem* [19] and *resynchronization problem* introduced by the IEEE 802.11p protocol [18]. The broadcast storm takes place whenever multiple vehicles attempt to transmit simultaneously, causing network overload, packet collisions and additional delay at the medium access control (MAC) layer. On the other hand, the resynchronization occurs due to multichannel operation of the IEEE 802.11p standard.

Aiming to provide a better understanding of the behaviour of FASTER, we present an extensive set of experiments that shows the need of novel solutions to improve vehicle traffic efficiency as well as clearly indicates its superior performance when compared to existing proposal.

This paper is organized as follows. Section 2 provides an overview of the existing TMS approaches and presents the related work. Section 3 introduce our proposed solution. The performance evaluation is described in Section 4 and, finally, Section 5 presents our conclusions and future work.

2. RELATED WORK

This section describes some related solutions to improve traffic efficiency found in the literature. However, is important to stress

MSWiM '16, November 13-17, 2016, Malta, Malta

© 2016 ACM. ISBN 978-1-4503-4502-6/16/11...$15.00

DOI: http://dx.doi.org/10.1145/2988287.2989167

that no fully-distributed solution aware about the overall traffic condition was found in the literature. There are some fully-distributed solutions which focus on control traffic jam caused by traffic accidents [8, 7]. Other solutions such as [1, 15, 2] focus on detect and control traffic jam cooperatively, but they did not have a full knowledge about the traffic condition. On the other hand, some centralized solutions uses a central entity to collect information periodically to create a this knowledge and to compute alternative routes for vehicles [6, 5]. Other solutions use a central entity to collect a full knowledge of traffic condition, but routes are computed in each vehicle, which may lead to not suitable alternative routes [1, 15] and overload the network [10, 6, 5].

Meneguette et al. [15] introduce the Urban CONgestion Detection System (UCONDES), a TMS based on inter-vehicle communication to detect and reduce urban congestion. UCONDES uses an Artificial Neural Network (ANN) to detect and classify the levels of congestion on roads. To identify and classify the traffic congestion, it uses the average speed and the density of vehicles on the road, which are periodically obtained via beacons sent by them all. After classifying the target road, the information is sent to the other vehicles via beacon messages. Upon receiving a congested road message, a vehicle decides if it keeps the current route or computes an alternative route. However, the mechanism to alert the vehicles about congested roads may overload the network since several vehicles in the same road can produce the same road classification and disseminate the same messages.

Bauza et al. [2] propose CoTEC (Cooperative Traffic congestion detECtion), a novel cooperative vehicular system based on V2V communications to detect traffic congestion using fuzzy logic. CoTEC uses CAM (Cooperative Awareness Messages) or beacon messages to periodically broadcast the road traffic condition. In addition, CoTEC uses fuzzy logic to detect a potential road traffic congestion locally at each vehicle. The fuzzy logic system was built based on Level-of-Service (LOS) present in Highway Capacity Manual (HCM) [4]. The LOS represents a quality measurement used to describe the operational conditions within a traffic flow. Therefore, when a traffic jam is detected, at first, each vehicle broadcasts its own estimation about the traffic jam and, then, with all estimations, vehicles collaboratively detect and characterize the road traffic congestion.

Similarly to CoTEC, Araujo et al. [1] propose CARTIM (Cooperative vehiculAR Traffic congestion Identification and Minimization) which is a proposal for collaborative identification and minimization of traffic congestion. Like CoTEC, CARTIM uses V2V communications to cooperatively measure the level of traffic congestion. CARTIM collects data from the vehicles (speed and density) periodically sent through beacons by all vehicles and with these information by using a fuzzy logic system it is able to measures the level of congestion. However, despite of CARTIM use a fuzzy logic system as well as CoTEC the rules were built using different metrics presented in HCM, thus CARTIM and CoTEC differ in the fuzzy logic rules and in the mechanism to spread the local traffic measurement through the network. Furthermore, when a traffic congestion is detected, CARTIM proposes a heuristic to change the vehicles route to minimize the traffic congestion detected.

Doolan and Muntean [10] introduce EcoTrec, a novel eco-friendly routing system for vehicular traffic that relies on V2V communication. EcoTrec, assumes that all vehicles are equipped with GPS receiver, tilt sensor and accelerometer, that gather information about position, angle, acceleration of the vehicle and road surface condition. Moreover, with these information, EcoTrec builds a *Vehicle Model* that is the vehicle behavior. Furthermore, every vehicle sends periodically beacon messages with its ID, position and speed, where this beacon is sent to the other vehicles. Upon receiving these beacon messages, every vehicle aggregates it with existing data and builds a *Traffic Model*. In addition, the messages are sent using *Endemic routing*, in order to make sure that the vehicles will receive it. Moreover, EcoTrec builds a *Road Model* that considers the traffic condition on the road. This *Road Model* are stored in a central server and is fed by the *Traffic Model* of each vehicle. Finally, EcoTrec computes the recommended routes for each vehicle. Its routing algorithm takes into account two main factors: *(i)* the road characteristics and; *(ii)* traffic condition, which are used to set the weights of each road and build a scenario overview. Thereafter, this scenario overview is sent to the vehicles through the VANET, and the vehicles are then routed according to the Dijkstra lowest edge weight algorithm. Each time that a vehicle receives new scenario overview, it re-computes its route.

The proposal mencioned above have some limitations such as: network overload [6, 5, 10] and no knowledge about traffic condition to reroute vehicles [1, 15, 10, 2, 16]. To overcome such limitations, we propose FASTER which is a fully-distributed TMS to improve the overall traffic efficiency that does not overloads the network nor introduces an undesire overhead for the system.

3. PROPOSED SOLUTION

This section describes FASTER, a fully-distributed VANET-based TMS to improve vehicle traffic efficiency. FASTER relies only on V2V communication to gather traffic information, detect congested roads and compute alternative routes and yet producing low overhead.

The main challenge is how to provide to all vehicles the full knowledge about the traffic condition of the entire scenario with low overhead. To this aim, in FASTER all vehicles share information with their neighbors, by doing that, each vehicle can increase its knowledge about the traffic condition. Furthermore, every vehicle shares its local knowledge with all vehicles. However, to overcome VANETs challenges, such as broadcast storm and the resynchronization. It segments the scenario in districts and proposes an aggregation mechanism to reduce the number of transmissions and a data dissemination protocol to incur in low overhead, short delays and large coverage. Finally, upon getting full knowledge of traffic condition each vehicle can detect congested roads and compute alternative routes to avoid them.

Consider a VANET environment where the road network can be modeled by 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), the set* $E = \{e_{01}, e_{12} \ldots, e_{ij}\}$ *corresponds to the set of roads (edges) which connects the intersections* $E \subseteq V \times V$*. Each edge* e_{ij} *is defined by a pair of subsequent vertices* $(v_i, v_j) \in V$*. Moreover,* $W = \{w_1, w_2, \ldots, w_i\}$ *is a set of weights representing the traffic condition, in which* $w : E \to \mathbb{R}^*_+$*. Let* $N = \{n_1, n_2, \ldots, n_i\}$ *be a set of vehicles (nodes) and* $\langle e_{ij}, \ldots, e_{mn} \rangle \mid e_{ij}, \ldots, e_{mn} \in E$ *be a route* $\forall n \in N$*. Furthermore, let* C *be the communication range of* $\forall n \in N$*.* In FASTER, each vehicle builds an subgraph $G' = (V', E')$, whose $V' \subseteq V$ is the subset of vertices covered by C, and $E' \subseteq E$ is the set of incident and out-going edges of each vertex in V'. Furthermore, each vehicle builds a local knowledge about the traffic condition E''. In this way, to provide a precise knowledge, each vehicle shares its local knowledge with all vehicles in the network. Finally with this precise knowledge built, each vehicle can detect a congested road e_{pq} verifying its weight w_{pq}, and compute an alternative route in case that it will pass through this road $\exists e_{pq} \in \langle e_{ij}, \ldots, e_{mn} \rangle$.

FASTER relies to four main phases: *A)* Information Gathering;

B) Information Aggregation; and *C)* Information Dissemination; and *D)* Congestion Detection and Avoidance which are described below.

3.1 Information Gathering

Each vehicle gathers information from its one hop neighbors to build a local knowledge about the traffic condition. For this, every vehicle provides information to its neighbors through *beacon messages*.

Beacon message: *is an IEEE 802.11p 2-layer periodic message sent by all vehicles in the Control Channel (CCH) every 2 seconds, which contains the current vehicle position* $e_{pq} \in \langle e_{ij}, \dots, e_{mn} \rangle \mid e_{pq}, e_{ij} \dots, e_{mn} \in E$ *and its average speed* s_{mean}.

Upon receiving beacon messages, each vehicle stores these messages received and in a predefined interval t, each vehicle creates a *Local knowledge about the traffic condition* E'', that have all roads e_{ij} within its coverage and their respective average speed $S_{e_{ij}}$. Additionally, these roads are classified as fully covered roads or partially covered roads, this classification is given by the function $\xi(n)$.

$$\xi(n) = \begin{cases} 1 & e_{ij} \in E' \\ 0 & \exists v \in e_{ij} \mid e_{ij} \notin E', v \in V' \end{cases}$$

where, 1 represents the case of fully covered road, in which an edge $e_{ij} \in E'$ is covered by vehicle n, and 0 represents the case of partially covered road, in which just one vertex $\exists v \in V'$ that composing an edge $e_{ij} \notin E'$ is covered by the vehicle n. Figure 1 shows an example of this classification, in which the road e_{ij} is fully covered and the road e_{jk} is partially covered.

Finally with all covered roads, at a predefined time t, each vehicle computes the average speed $S_{e_{ij}}$ $\forall e \in E''$ based on associated speed s_{mean} of each *beacon message* received related to that road. However, in case that a vehicle covers a road e_{ij} and it does not receives any beacon related to that road, it sets the average speed $S_{e_{ij}}$ of that road with the road *max speed*, because it means that the road is free.

3.2 Information Aggregation

FASTER segments the scenario in districts and employs an aggregation mechanism to aggregate the *local knowledge about the traffic condition* of all vehicles in the same district in order to create a knowledge about the traffic condition of the district. To define the number of districts, FASTER uses Equation 1.

$$\kappa = \left\lceil \frac{A}{\pi \cdot C^2} \right\rceil \quad (1)$$

where, A is the scenario area, C is the vehicle communication range and κ is the number of districts.

Upon defining the number of districts, FASTER uses the kmeans clustering algorithm [14] to segment the κ districts. The key idea is to define κ centroids, one for each district, then associate a set of nearest vertices of a centroid to each district. Therefore a set of κ vertices $v_0, \dots, v_\kappa \mid v_0, \dots, v_\kappa \in V$ are selected to represent the centroids, then every vertex is bind to nearest centroid, forming a district. Furthermore the κ centroids are re-calculated and a new bind is done. This process is employed until the κ centroids stay the same. The district selecting process aims at minimizing an objective function, in this case a squared error function. The objective function is defined by Equation 2.

$$\underset{V}{\operatorname{argmin}} \sum_{j=1}^{\kappa} \sum_{v \in V_j} \| v - c_j \|^2 \quad (2)$$

where, $\| v - c_j \|^2$ is the distance measure between vertex v and the district centre c_j, V_j is the set of vertices associated to the district j and κ is the number of districts.

Figure 1 shows an example of district segmentation, and its centroids to a scenario with an area of $1km^2$ and vehicles with communication range of 300 m.

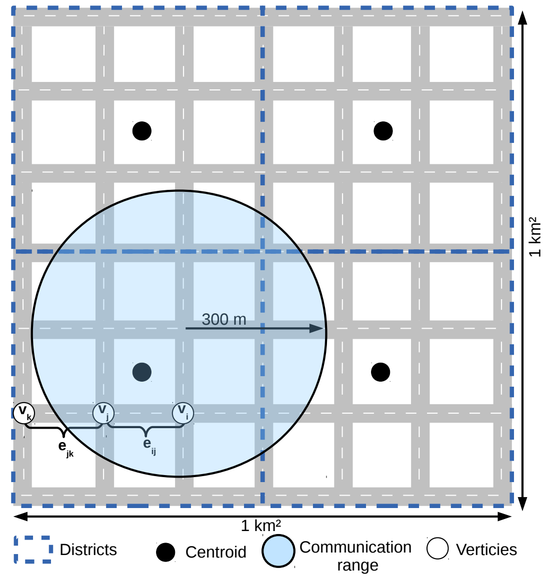


Figure 1: District organization.

After each vehicle creates its local knowledge of traffic condition, a *Local knowledge message* is created to share its local knowledge about the traffic condition with all one hop neighbors in the same district.

Local knowledge message: *is an IEEE 802.11p 2-layer periodic message sent by all vehicles in the Service Channel (SCH) in an interval* $t + \gamma$, *which contains all covered roads* $e_{ij} \in E''$, *the average road speed* $S_{e_{ij}}$ *and the road type* $[0, 1]$, *where* 0 *is to partially covered roads and* 1 *to fully covered roads.*

When a vehicle n_i receives a *Local knowledge messages* from a vehicle n_j it creates a *District knowledge*, which have all roads and their average speed in relation to its district. Algorithm 1 shows the creation of a district knowledge.

In this way, for each entry of each *Local knowledge message* received from the vehicle n_j. First, the vehicle n_i verifies if it already knows the traffic condition of the that road (Line 4). To this, it verifies if it already has the road e_{ij} in its *District knowledge*. If it does not knows the traffic condition of the road e_{ij} received from vehicle n_j, it adds the road to its *District knowledge* (Lines 5-6). On the other hand, if it already knows the traffic condition of the road e_{ij}, it verifies the road type (Line 9) and in case of the road in its *District knowledge* is fully covered, the vehicle keeps its local knowledge about the traffic condition of that road. Otherwise in the case that the road in its *District knowledge* is partially covered and the road in the *Local knowledge message* received is full covered, its updates its knowledge of the road traffic condition (Lines 10-12). However if both roads (the road which the vehicle n_i has in its District knowledge and the road received from the local knowledge of vehicle n_j) have type partially covered, vehicle n_i updates its *District knowledge* using the minimum of the both roads average speed (Line 15).

Despite of each vehicle has knowledge about the traffic condition in its *District knowledge*, this knowledge is far from of the full

Algorithm 1: District knowledge creation

Input : E''_{n_i} `// Set of all covered roads of vehicle` n_i
E''_{n_j} `// Local knowledge received from vehicle` n_j

Output: District knowledge built by the vehicle n_i, which contains each district road e_{ij} and its average speed $S_{e_{ij}}$

`// set the local knowledge of vehicle` n_i `to its district knowledge` D
$D \leftarrow E''_{n_i}$;
foreach $e_{ij} \in E''_{n_j}$ **do**
    `// verifies if the edge` e_{ij} `already is in its district knowledge`
    if $e_{ij} \notin D$ **then**
        `// vehicle` n_i `adds the edge` e_{ij}`, average speed` $S_{e_{ij}}$ `and the edge type received from vehicle` n_j `to its district knowledge`
        $D[e_{ij}].S_{e_{ij}} \leftarrow E''_{n_j}[e_{ij}].S_{e_{ij}}$;
        $D[e_{ij}].type \leftarrow E''_{n_j}[e_{ij}].type$;
    end
    else
        `// verifies if the edge` e_{ij} `in its district knowledge has type partially covered`
        if $D[e_{ij}].type$ *is partially covered* **then**
            `// verifies if the edge` e_{ij} `received from vehicle` n_j `is full covered`
            if *if* $E''_{n_j}[e_{ij}].type$ *is fully covered* **then**
                `// updates the edge speed and the type`
                $D[e_{ij}].S_{e_{ij}} \leftarrow E''_{n_j}[e_{ij}].S_{e_{ij}}$;
                $D[e_{ij}].type \leftarrow 1$;
            end
            else
                `// updates the average speed`
                $D[e_{ij}].S_{e_{ij}} \leftarrow \min(D[e_{ij}].S_{e_{ij}}, E''_{n_j}[e_{ij}].S_{e_{ij}})$
            end
        end
    end
end

scenario knowledge desired. Therefore to have such full knowledge, each vehicle needs to know the *District knowledge* of all κ districts. In other words, each vehicle needs to have knowledge about the traffic condition of the others districts. To this, FASTER implements an inter-district data dissemination protocol, which is explained below, to spread each *District knowledge* to other districts.

3.3 Information Dissemination

FASTER employs a multi hop delay-based data dissemination protocol, which is composed by three main modules: *(i)* Transmitter selection; *(ii)* Broadcast suppression; and *(iii)* Packet collision minimization.

Due to aggregation mechanism employed, all vehicles within the same district potentially share the same *District knowledge*. In this way, a single transmission of this knowledge is sufficient to forward it to the others districts. In order to provide such issue, FASTER uses its *Transmitter selection* module, to select the best vehicle to transmit its *District knowledge*. In a predefined interval $t + \lambda \mid \lambda > \gamma$, all vehicles create a *District knowledge message*, which contains its *District knowledge*. A delay is computed to schedule this message based on the euclidean distance between its current position and the district centre. The main idea of *Transmitter selection* is to prioritize the transmission of the vehicle nearest to the center. The vehicle closest to the center will transmit first then the others; when the other vehicles receive a *District knowledge message* from its district, they cancel their *District knowledge message* scheduled. Moreover the *District knowledge message* needs to be forward to the other districts by employing a multi-hop process starting the *Broadcast suppression* module.

The *Broadcast suppression* module addresses the broadcast storm problem by reducing the number of transmissions, providing different delays and canceling duplicated messages. To this, is used a *preference zone* concept [20], which is an area where the vehicles are best suited to continue performing forwarding. In other words, among all vehicles that received data to be forwarded, the transmission of a single vehicle within the *preference zone* is sufficient to perform the data dissemination efficiently. Therefore, vehicles within a *preference zone* have a lower delay then the vehicles outside this area. Figure 2 shows the *preference zone* organization to a transmitter vehicle, and the respective delays for vehicles within the *preference zone* and as well as outside this zone.

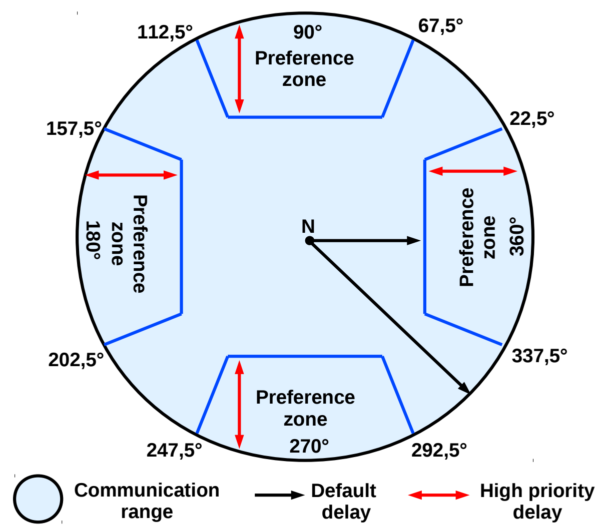


Figure 2: Preference zone organization.

FASTER also employs a desynchronization mechanism to avoid the *synchronization problem* introduced by the IEEE 802.11p standard [11, 18]. The desynchronization mechanism proposed aims to reduce packet collision. For this, it verifies if the delay computed by the *Broadcast suppression* module will occur when the service channel becomes active. In case this delay happens, the *desynchronization mechanism* adds an additional delay to allow transmission in the service channel, where the additional delay at most 50 ms (the time the IEEE 802.11p standard uses to swap from CCH to SCH). If a transmission is scheduled when the control channel is active, the additional delay is added to allow the transmission to occur during the SCH. Figure 3 shows how the desynchronization mechanism works. In this figure, a vehicle receives a *District knowledge message* at time T_1, and the *Broadcast suppression* computes a delay to forward the *District knowledge message* in 60 ms. After 60 ms, at time T_2 the CCH will be active, avoiding the synchronization, then the desynchronize mechanism computes again the delay and changes it to 115 ms.

3.4 Congestion Detection and Avoidance

After a vehicle receive all κ *District knowledge*, at time $t + \rho \mid \rho > \lambda$, it creates a *Scenario knowledge*, which contains all roads and average speed received from all κ districts. Each vehicle builds

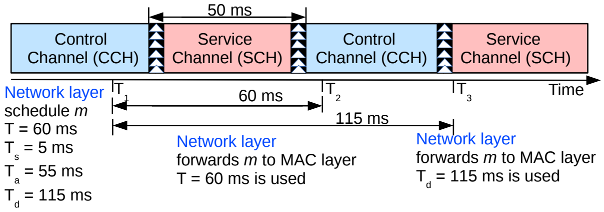


Figure 3: Desynchronized meachism.

Table 1: Relation of weight, LOS and traffic classification

Road weight	LOS	Traffic classification
$(0, 0.15]$	A	Free-flow
$(0.15, 0.33]$	B	Free-flow
$(0.33, 0.50]$	C	Sligth congestion
$(0.50, 0.60]$	D	Sligth congestion
$(0.60, 0.70]$	E	Congested
$(0.70, 1.00]$	F	Congested

a graph $G = (V, E)$ based on *Scenario knowledge* and for each road $e_{ij} \in E$ it computes a weight w_{ij} based on Equation 3. The weight equation was modeled to be inversely proportional to the road average speed value. In this way, faster roads have lower weights then slower roads, as follows:

$$w_{ij} = 1 - \frac{S_{e_{ij}}}{S_{e_{ij}}^{max}} \tag{3}$$

where $S_{e_{ij}}$ and $S_{e_{ij}}^{max}$ denote the max speed, and the average speed in *Scenario knowledge* of road e_{ij}, respectively.

Each vehicle can classify the traffic condition in each road e_{ij} based on the each weight w_{ij} value. The *Congestion detection* module is based on the Level-Of-Service (LOS) of the Highway Capacity Manual (HCM) [4]. The LOS represents a quality measurement used to describe the operational conditions within a traffic flow, which defines six different levels of service, with LOS A representing free-flow conditions and LOS F representing high congested level. Furthermore each level defines the minimum and the maximum speed based on the road maximum speed. Table 1 shows the minimum and maximum road weight to each LOS from HCM.

The *Congestion detection and avoidance* module is divided in two phases: *(i)* Real-time decision making, which each vehicle verifies if it will pass through a congested road and *(ii)* Alternative route computation, which relies to compute an alternative route if the vehicles will pass through a congested road. In this context, relating to *Real-time decision making* phase, to enable a detection of a congested road, each vehicle verifies all weights of the roads that compose its route, and if it has a road with a weight greater than a predefined threshold it knows that the road is congested. The a threshold is defined based on the HCM manual with the value of 0.50 (LOS D). The *Real-time decision making* function $\delta(n)$ is given by.

$$\delta(n) = \begin{cases} 1 & \exists e_{pq} \in \langle e_{ij}, \ldots, e_{mn} \rangle \mid w_{pq} \geq 0.50, e_{pq} \in E \\ 0 & Otherwise \end{cases}$$

The value 1 means that a vehicle has road e_{pq} in its route $\langle e_{ij}, \ldots, e_{mn} \rangle$ which has weight w_{pq} up to the defined threshold and 0 otherwise.

In case that a vehicle will go through a congested road, the *Alternative route computation* phase starts. In this way, the vehicle computes an alternative route using the Probabilistic k-Shortest Path (PkSP) based on the road weights. PkSP uses the Boltzmann probabilistic algorithm [13] to select the route to the vehicle among the ones in a set of *k* possible routes.

4. PERFORMANCE ANALYSIS

This section describes the performance evaluation of FASTER. First of all, in Subsection 4.1 the simulation tools is presented as well as the scenario used in our simulations. Subsections 4.2 and 4.3 present the network cost and traffic efficiency results respectively. The results were organized as the following. First we analysed the network cost of our solution by evaluating the aggregation mechanism and the data dissemination protocol. Second, we compared FASTER with some TMS solutions to evaluate the traffic efficiency.

4.1 Methodology

For the simulations, we use the Network Simulator OMNeT++ 4.3[1] and the framework SUMO (Simulator for UrbanMObility) [3], version 0.25.0, to manage the scenario and the mobility of vehicles. We obtained realistic map scenarios from OpenStreetMap tool. For the vehicular network, we used the framework Veins 4.3 [17] that implements the IEEE 802.11p and the signal attenuation model considering obstacles.

We used as scenario a fragment (1 km^2) of Manhattan, USA, with several blocks and two-way streets for vehicles moving in opposite directions. The route of the vehicles is composed by the shortest path between two vertices $v_i, v_j \in V$ which are selected randomly. Furthermore to ensure close to real overtaking, their speeds range from 0 kilometres per hour to the maximum speed allowed on the road. In addition, to provide different traffic conditions, we used densties from 700 to 1500 vehicles/km^2. In this way, as main parameters, we set the bit rate to 18 Mbit/s at the MAC layer and the transmission power to 2.2 mW, resulting in a communication range of approximately 300 m under a two-ray ground propagation model [17]. Table 2 shows a summary of the main simulation parameters used in our assessment.

Table 2: Simulation parameters

Parameters	Values
Transmission power	2.2 mW
Communication range	300 m
Bit rate	18 Mbit/s
Scenario	Manhattan grid
Scenario size	1 km^2
Densities	from 700 to 1500 vehicles/km^2
Number of simulations	33
Confidence interval	95%

4.2 Network Cost evaluation

As there is no other fully-distributed solution with the same purpose of FASTER in literature, we evaluate the network cost comparing with the Baseline, in which all vehicles disseminate their local knowledge using the data dissemination protocol proposed with no aggregation. The assessed metrics for the network cost evaluation were:

- **Coverage**: the percentage of messages generated by the vehicles regarding to the delivered messages;

[1] http://omnetpp.org/

Table 3: Coverage

Vehicles/Km^2	Baseline	FASTER
700	61.13%	93.39%
900	57.49%	95.11%
1100	51.31%	95.26%
1300	51.78%	94.63%
1500	45.27%	95.18%

Table 4: Average delay (s)

Vehicles/Km^2	Baseline	FASTER
700	114.46	0.44
900	102.88	0.45
1100	137.81	0.47
1300	288.58	0.48
1500	533.20	0.49

Table 5: Transmitted messages

Vehicles/Km^2	Baseline	FASTER
700	2910	179
900	4834	206
1100	6864	216
1300	10201	246
1500	13833	279

Table 6: Overall traffic knowledge

Vehicles/Km^2	Baseline	FASTER
700	62.63%	97.58%
900	49.37%	95.63%
1100	40.53%	96.23%
1300	31.13%	95.82%
1500	12.66%	95.6%

- **Delay**: the average time to spread district knowledge messages to all vehicles;
- **Transmitted messages**: total number of district knowledge messages transmitted;
- **Overall traffic knowledge**: accuracy of the knowledge about the traffic condition that each vehicle have about the traffic condition of the entire scenario.

Tables 3, 4, 5 and 6 show results for all assessed networks metrics in function of the vehicle density. Table 3 shows the results for coverage. In particular, FASTER reaches a coverage of approximately 95% when the density of 700 vehicles/km^2, due its efficient aggregation mechanism that aggregates all knowledges of each district in a single one, thus reducing the number of transmitted messages. On the other hand,the Baseline presents a coverage of approximately 60%. Its low coverage presented occurs because it does not implements any mechanism to aggregate messages in order to reduce the number of transmissions, consequently as all vehicles need to transmit their local knowledge the number of transmitted messages increase and overloads the network. However, this overload worsens as the density increases. Therefore, with the density of 1500 vehicles/km^2 the Baseline system presents a coverage of only 45%, because it is not able to deliver all messages transmitted by all vehicles. Meanwhile, FASTER maintains its coverage of 95%. It is effect of the efficient aggregation mechanism employed by FASTER, which reduce the number transmitted messages (see Table 5) and consequently it does not overload the network.

Overloading the network creates several negative impacts for the system, it introduces an undesired overhead that affects its overall performance. In Table 4 is shown the average delay that each system introduces to spread the knowledge messages through the network. As can be expected, the overload caused by the Baseline system introduces a high overhead which presents a delay greater than 100 seconds for all densities. This overhead degrades the overall performance of the system, in which in scenarios with frequent traffic hazards such as hush hours, that a fast response is desired to provide information about the traffic condition for the vehicles, it may take almost 10 minutes to spread the knowledge, and also it may not be a precise one (see Table 6), thus infeasible to provide a suitable solution. Differently, FASTER does not overloads the network, therefore it is able to spread the knowledge through the network in a fast way, specifically it takes less than 1 second. Such results in FASTER is consequence of the adoption of the efficient aggregation mechanism, which drastically reduces the amount of transmitted messages.

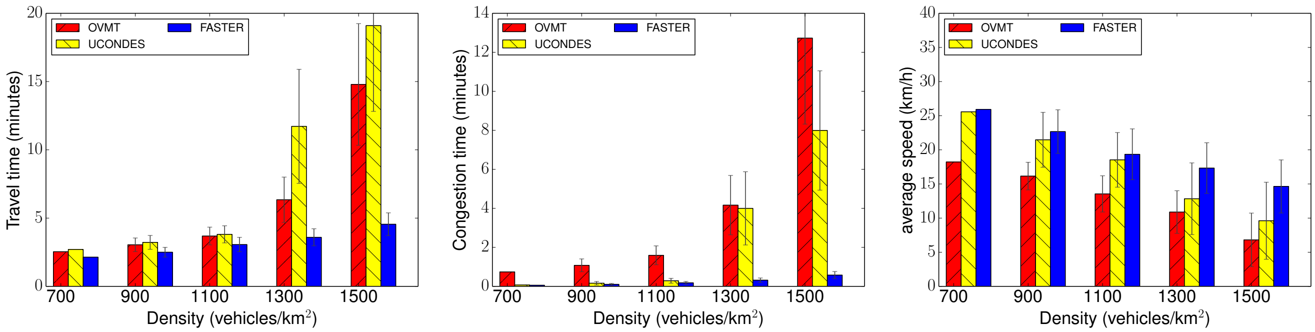
Table 5 shows the total number messages transmitted in each system for all densities. As discussed above, the Baseline system produces a high number of messages because it does not implements an aggregation mechanism, so all vehicles need to transmit their messages at least once. However, this approach transmits several of redundant messages which could be aggregate in a single one to make a better usage of the network. This feature, is implemented in FASTER, and as can be seen it generates approximately 97% less messages when compared to Baseline. Such result shows that FASTER does not waste bandwidth with unnecessary transmissions.

Finally, the impact of the network overload present in Baseline system can be seen in Table 6, which presents the overall knowledge that each vehicle has about the traffic condition of the entire scenario. The Baseline system, starts with an accuracy of 62% with the density of 700 vehicles/km^2. Despite this is not a good accuracy, it get worse as the density increases, because more messages are transmitted, consequently this further overloads the network and increases the overhead. Consequently, Baseline presents an accuracy of 12% with the density of 1500 vehicles/km^2. However, thanks for the efficient mechanisms employed by FASTER, it presents an accuracy between 98% and 95% for all densities. This results shows how efficient is FASTER to spread the information in order to build a precise knowledge about the traffic condition of the entire scenario.

4.3 Traffic Efficiency evaluation

In this section, we evaluate the efficiency of vehicular traffic flow using FASTER and UCONDES [15], compared with the original vehicle mobility traffic (OVMT). The assessed metrics for the vehicular traffic efficiency were: *(i)* Travel time; *(ii)* Congestion time; and *(iii)* Average speed. Regarding FASTER parameters, we used $t = 15$ seconds, $\gamma = 1$ second, $\lambda = 1$ second, and $\rho = 2$ seconds.

Figure 4 shows the results for all vehicular traffic efficiency metrics. In particular, Figure 4(a) shows the travel time results. It is important to note that OVMT represents the original mobility (i.e. it does not alternates any route of all vehicles). Therefore, as can be seen the more the density increases, the travel time increases as well, reaching approximately 15 minutes with density of 1500 vehicles/km^2. However, UCONDES presents the highest travel time increasing up to 30% compared to OVMT. This is consequence of the absence of the knowledge about the traffic condition



(a) Travel time (b) Congestion time (c) Average speed

Figure 4: Traffic efficiency evaluation results.

in the entire scenario. Vehicles in UCONDES only know about the roads that are congested within its coverage. Therefore, they compute an alternative route to avoid the congested roads just based on the distance between its current position and its destination. Hence, they may get stuck in another congestion (see Figure 4(b)). On the other hand, as vehicles in FASTER have an overall knowledge about the traffic condition, they can compute improved alternative routes to avoid congested roads avoiding to enter in another congestion. Moreover, it reduces the average travel time in 58% and 68% compared with OVMT and UCONDES respectively.

Regarding congestion time, in Figure 4(b) OVMT presents the higher congestion time. Vehicles spend up to 12 minutes of their entire travel time in the congestion. In other words, vehicles spend up to 80% of their entire route stuck in some congestion. UCONDES reduces this time spent in some congestion to approximately 7 minutes, reducing the congestion time in up to 45% compared to OVMT. However, as it does not compute efficient routes, the vehicles still spend 33% of its route stuck in some congestion. On the other hand, FASTER presents the lower congestion time, which is reduce to approximately 1 minute. it reduces the congestion time in 93% and 87% compared to OVMT and UCONDES. FASTER reaches better results because all vehicles have a precise knowledge about the traffic condition of the entire scenario. In addition, it is possible because it does not overload the network to provide such knowledge using the efficient aggregation and the data dissemination mechanisms, consequently it does not introduces a higher overhead for the system making possible that vehicles detect and avoid traffic hazards in a fast way. Finally, as consequence of the lower travel and congestion time FASTER enables a smooth traffic flow increasing the average speed in up to 90% when compared to OVMT and 50% when compared to UCONDES.

The better traffic efficiency presented in FASTER is consequence of low overhead and the high accuracy of the overall traffic knowledge, thus it can detect congestion first than UCONDES [15]. Such low overhead is because FASTER is fully-distributed, where each vehicle can detect a traffic hazard and compute an alternative to itself. Differently form centralized systems which concentrate the calculation of an alternative route to all vehicles in a single entity, in which can introduce high overhead depending on the traffic density.

5. CONCLUSION

Vehicle congestion has become a daily problem affecting several aspects of modern society. Nowadays it is extremely necessary a distributed TMS to improve the flow of vehicle traffic once the centralized TMS present several issues. In this work we present the FASTER, a fully-distributed VANET-based TMS to improve the overall vehicle traffic efficiency as well as to avoid traffic congestion. FASTER was extensively compared to other known solutions regarding travel time, congestion time, and average speed and several network metrics. The obtained results indicate that our FASTER outperforms the assessed solutions in different scenarios and in different key requirements of TMS. As future work, we intend to analyze FASTER in more realistic scenarios, using real environments mobility traces. In addition, we intend to extend our solution implementing more sophisticated re-reroute algorithms.

6. 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.

7. REFERENCES

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

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

[3] M. Behrisch, L. Bieker, J. Erdmann, and D. Krajzewicz. SUMO - Simulation of Urban MObility: An Overview. In *International Conference on Advances in System Simulation (SIMUL '11)*, pages 63–68, 2011.

[4] T. R. Board. *HCM 2010 - Highway capacity manual.* National Research Council, 2010.

[5] A. M. de Souza, R. S. Yokoyama, L. C. Botega, R. I. Meneguette, and 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*, pages 497–503, Oct 2015.

[6] A. M. de Souza, R. S. Yokoyama, G. Maia, A. Loureiro, and L. Villas. Real-time path planning to prevent traffic jam through an intelligent transportation system. In *2016 IEEE Symposium on Computers and Communication (ISCC)*, pages 726–731, June 2016.

[7] A. M. de Souza, R. S. Yokoyama, G. Maia, A. A. F. Loureiro, and 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)*, pages 1–5, July 2015.

[8] A. M. de Souza R. S. Yokoyama Nelson L. S. da Fonseca Rodolfo I. Meneguette and L. A. Villas. Garuda: A new geographical accident aware solution to reduce urban congestion. In *Proceedings of the 15th IEEE International Conference on Computer and Information Techinology*, 2015.

[9] S. Djahel, R. Doolan, G.-M. Muntean, and J. Murphy. A Communications-Oriented Perspective on Traffic Management Systems for Smart Cities: Challenges and Innovative Approaches. *IEEE Communications Surveys Tutorials*, 17(1):125–151, 2015.

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

[11] 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*, pages 1–5, May 2012.

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

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

[14] J. B. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability*, 1967.

[15] R. I. Meneguette, J. Ueyama, G. P. R. Filho, B. Krishnamachari, L. F. Bittencourt, and L. A. Villas. Enhancing intelligence in inter-vehicle communications to detect and reduce congestion in urban centers. In *IEEE Symposium on Computers and Communication (ISCC)*, 2015.

[16] J. Pan, I. S. Popa, and C. Borcea. Divert: A distributed vehicular traffic re-routing system for congestion avoidance. *IEEE Transactions on Mobile Computing*, PP(99):1–1, 2016.

[17] C. Sommer, R. German, and F. Dressler. Bidirectionally coupled network and road traffic simulation for improved ivc analysis. *Mobile Computing, IEEE Transactions on*, 10(1):3–15, Jan 2011.

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

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

[20] 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*, 2014.