HAL
open science

Socially Inspired Data Dissemination for Vehicular Ad Hoc Networks

Felipe Domingos da Cunha, Guilherme Maia, Aline Carneiro Viana, Raquel A. F. Mini, Leandro Villas, Antonio A. F. Loureiro

To cite this version:

Felipe Domingos da Cunha, Guilherme Maia, Aline Carneiro Viana, Raquel A. F. Mini, Leandro Villas, et al.. Socially Inspired Data Dissemination for Vehicular Ad Hoc Networks. MSWiM '14 Proceedings of the 17th ACM international conference on Modeling, analysis and simulation of wireless and mobile systems, Sep 2014, Montreal, Canada. 10.1145/2641798.2641834 . hal-01082984v1

HAL Id: hal-01082984

https://inria.hal.science/hal-01082984v1

Submitted on 14 Nov 2014 (v1), last revised 24 Sep 2015 (v2)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Socially Inspired Data Dissemination for Vehicular Ad Hoc Networks*

Felipe D. Cunha
DCC - UFMG
fdcunha@dcc.ufmg.br

Guilherme Maia
DCC - UFMG
jgmm@dcc.ufmg.br

Aline C. Viana
INRIA - Saclay
aline.viana@inria.fr

Raquel A. F. Mini
DCC - PUC Minas
raquelmini@pucminas.br

Leandro A. Villas
IC - Unicamp
leandro@ic.unicamp.br

Antonio A. F. Loureiro
DCC - UFMG
loureiro@dcc.ufmg.br

ABSTRACT

People have routines and their mobility patterns vary during the day, which have a direct impact on vehicular mobility. Therefore, protocols and applications designed for Vehicular Ad Hoc Networks need to adapt to these routines in order to provide better services. With this issue in mind, in this work, we propose a data dissemination solution for these networks that considers the daily road traffic variation of large cities and the relationship among vehicles. The focus of our approach is to select the best vehicles to rebroadcast data messages according to social metrics, in particular, the clustering coefficient and the node degree. Moreover, our solution is designed in such a way that it is completely independent of the perceived road traffic density. Simulation results show that, when compared to related protocols, our proposal provides better delivery guarantees, reduces the network overhead and possesses an acceptable delay.

Categories and Subject Descriptors

C.2.0 [**Computer-Communications Networks**]: General—*Data communications*; C.2.1 [**Computer-Communications Networks**]: Network Architecture and Design—*Wireless communication*

Keywords

VANETs; Broadcast Suppression; Intermittently Connected; Social Metrics

1. INTRODUCTION

Vehicular Ad Hoc Networks (VANETs) are a special type of Mobile Ad Hoc Networks in which vehicles have processing and wireless communication capabilities. Usually, these vehicles exchange information among themselves through multi-hop communication. In these networks, sending messages from a source to all vehicles located inside a geographic region will be very common. Such activity is known as data dissemination. Data dissemination solutions must consider two important challenges. The first one, known as the *broadcast storm problem*, happens when a group of vehicles close to one another starts to transmit data messages at the same time, leading to a high number of message collisions and severe contention at the link layer [7, 9]. The second one, known as the *intermittently connected network problem*, happens in scenarios with low traffic densities, such as, daybreak, holidays and rural areas, in which the number of vehicles is not enough to disseminate data messages using direct multi-hop communication [6, 11].

A factor that contributes to the emergence of these problems is the driver's routine. Usually, people posses similar behavior, which increases the likelihood of going to the same places at the same time. Moreover, while moving around, drivers are susceptible to speed limits, traffic lights, obstacles, etc. Therefore, it is reasonable to assume that these factors combined lead to microscopic and macroscopic traffic density variations. We argue that, a better understanding of these routines and their impact on the overall traffic condition is fundamental in designing better communication protocols for VANETs.

There is a vast literature that investigates the social aspects inherent to VANETs [2–4]. In summary, they show that there are social properties encoded in these networks. With this in mind, in this work, we leverage these social aspects to design a *Socially Inspired Broadcast Data Dissemination* for VANETs. In our approach, we use two social metrics, *clustering coefficient* and *node degree*, to determine when vehicles should rebroadcast data messages in order to increase the delivery guarantee and reduce the overall network overhead, independently of the perceived road traffic condition. Simulation results show that, when compared to two related protocols – UV-CAST [11] and ABSM [6] – under a Manhattan grid scenario, our solution possesses a higher delivery ratio, decreases the total number of data messages transmitted and the number of collisions, and it also has an acceptable delay.

The remaining of this paper is organized as follows. Section 2 summarizes the recent related work. Section 3 presents our socially inspired proposal. Section 4 describes the simulation scenarios and discusses the results. Finally, Section 5 presents our final remarks and some future work.

2. RELATED WORK

Due to the peculiarities of VANETs, such as: vehicles move at very high speeds, frequent topology changes and short encounter times, the traditional solutions do not present good performance. Thus, many solutions especially designed for this network have been proposed [6, 7, 9, 11]. For instance, Ros et al. [6] propose the ABSM, a data dissemination protocol for VANETs with varying road traffic conditions. The key idea of ABSM is to use the Minimum Con-

*This work has been partially supported by INRIA, Fapemig and CNPq.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
MSWiM'14, September 21–26, 2014, Montreal, QC, Canada.
Copyright 2014 ACM 978-1-4503-3030-5/14/09 ...$15.00.
http://dx.doi.org/10.1145/2641798.2641834.

nected Dominating Set (MCDS) concept. In summary, if only the vehicles in the MCDS rebroadcast a data message, then 100% coverage is guaranteed with a low overhead. However, determining the MCDS is a NP-hard problem. Therefore, the authors outline a distributed heuristic to determine whether a vehicle belongs to the MCDS or not. Vehicles in the MCDS are assigned a lower waiting delay to rebroadcast. To guarantee message delivery under intermittently connected VANETs, ABSM relies on periodic beacons as implicit acknowledgements. For that, vehicles insert the IDs of received data messages into beacons. When a vehicle receives a beacon from a neighbor and it does not acknowledge the receipt of a message, then the vehicle forwards the message to the neighbor.

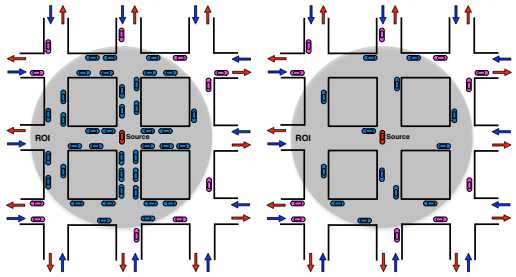
Viriyasitavat et al. [11] propose UV-CAST for performing data dissemination under both dense and sparse VANETs. In UV-CAST, a vehicle can be in one of two states, broadcast suppression or store-carry-forward. When a vehicle receives a data message for the first time, it initially checks whether it is a border vehicle or not. Border vehicles are the ones that are at the edge of a connected component. UV-CAST assumes these vehicles have a higher probability of meeting new neighbors. If the vehicle verifies it is a border vehicle, then it stores the message and carries it around until an encounter with a new neighbor is made. Conversely, if the vehicle is not a border vehicle, it executes a broadcast suppression algorithm to rebroadcast the message.

3. PROPOSED SOLUTION

In this work, we choose to apply the clustering coefficient and node degree to design an efficient data dissemination solution for VANETs. We focus on these two metrics because they provide the possibility to be aware of vehicles density in a region and consequently, to adjust the dissemination in an efficient way. With the information extracted from those algorithms, our solution find network nodes that behaves as hub nodes (or start nodes): Nodes that have a lot of neighbors (i.e., have high degree) that do not see each other (i.e., have low cluster coefficient). These are very good candidates to guarantee the best performance in higher density regions and reach sparse regions with a low cost. This is possible because the metrics computation just use the beacons packets exchanged among the vehicles, in other words, packets that are already exchanged for others purposes in the network.

Data dissemination corresponds to the process in which a single source vehicle or roadside unit broadcasts data messages to all vehicles located inside a region of interest (ROI) through multi-hop communications, as illustrated in Figure 1. The ROI is defined by the application for which the messages must be disseminated. Moreover, in this work, we assume the ROI is defined as a circular region centered at the source. The main goal of our proposal is to guarantee message delivery to all vehicles inside the ROI independently of the road traffic condition. Therefore, the protocol must be able to operate under both dense (Figure 1(a)) and sparse (Figure 1(b)) VANETs, and for that, both the broadcast storm and intermittently connected network problems must be tackled.

We assume that vehicles store and carry each received data message for the whole period in which they are inside the ROI and the time-to-live for the message has not expired. Moreover, they are equipped with a Global Positioning System (GPS) or they can infer their positions through other means. Each vehicle periodically exchanges beacons with its neighbors. These beacons contain context information about the vehicle, for instance, the position and the number of neighbors (node degree). Furthermore, each beacon contains the IDs of the data messages which have being received and are being carried by the vehicle. Notice that, embedding the IDs of received data messages into beacons works as an implicit acknowledgement mechanism. Therefore, when a vehicle receives a



(a) Dense traffic (b) Sparse traffic

Figure 1: Data dissemination to a group of vehicles under both dense and sparse traffic scenarios

beacon from a neighbor, it is able to verify whether it possesses any data message that has not been received by this neighbor and then, forward it accordingly.

Algorithms 1 and 2 show the main steps of our proposed data dissemination solution for both dense and sparse road traffic scenarios. Essentially, what these algorithms do is to determine which vehicles should rebroadcast a received data message and when they should perform it. By carefully coordinating these tasks, our proposal is able to avoid redundant retransmissions and increase the delivery probability to intended recipients. In the following two sections, we thoroughly describe each algorithm. Thereafter, we show how the clustering coefficient and node degree is used to calculate the waiting delay for a vehicle to rebroadcast.

3.1 Broadcast Suppression

Under dense road traffic conditions, when a vehicle receives a data message, it must carefully decide whether to rebroadcast it or not, and when to rebroadcast it in order to avoid redundant retransmissions and consequently, the broadcast storm problem. Algorithm 1 shows how a vehicle proceeds when it receives a data message m.

Algorithm 1: The broadcast suppression algorithm

1 **Event** *data message m received from neighbor s*
2 **if** *vehicle is outside the region of interest specified in m or the time-to-live of m expired* **then**
3 discard m;
4 **if** *m is not a duplicate* **then**
5 add message to the list of received messages;
6 insert m ID in subsequent beacons;
7 $t \leftarrow$ calculateWaitingDelay();
8 schedule $rebroadcast_timer$ for m to fire up at $currentTime + t$;
9 **else**
10 **if** *$rebroadcast_timer$ for m is scheduled* **then**
11 cancel $rebroadcast_timer$ for m;
12 **Event** *scheduled $rebroadcast_timer$ for m expires*
13 Rebroadcast m;

Initially, the vehicle verifies whether it has left the ROI or the time-to-live for the message m has expired. In such case, the vehicle discards m (lines 2–3). Otherwise, the vehicle checks whether m is a duplicate or not (Line 4). If it is not a duplicate, then the vehicle stores m in the list of received messages that are still valid. Furthermore, it will insert the ID of m into subsequent beacons,

until the vehicle leaves the ROI or m expires (lines 5–6). The next and most important step is to calculate the waiting delay t to rebroadcast m (Line 7). In Algorithm 1, we omitted how this delay is calculated, because it will depend of the social metric employed, i.e., the clustering coefficient, node degree or both, as described in Section 3.3. For now, it is enough to know that such delay is a value in the interval [0, T_{max}], where T_{max} is a configured parameter. After calculating the waiting delay, the vehicle uses it to schedule a rebroadcast for m (Line 8). Notice that, while the vehicle is scheduled to rebroadcast m, if it receives a duplicate, then it cancels the rebroadcast (lines 9–11), thus avoiding a possible redundant retransmission. However, when the waiting delay expires and the vehicle has not received any duplicate, then it rebroadcasts m (lines 12–13).

3.2 Store-carry-forward

On the other hand, when the road traffic is sparse and the network is partitioned, vehicles must hold received data messages and use their mobility capabilities to carry the messages to different parts of the ROI. Moreover, they must be able to determine whether a vehicle has already received a data message or not. For the former issue, vehicles rely on the store-carry-forward communication model. For the latter, beacons are used as an implicit acknowledgement mechanism. Algorithm 2 shows how our proposed solution delivers data messages even when the network is intermittently connected.

Algorithm 2: The store-carry-forward algorithm

```
1  Event beacon b received from neighbor s
2  foreach message m in the list of received messages do
3      if m is not acknowledged in b then
4          t ← calculateWaitingDelay();
5          schedule rebroadcast_timer for m to fire up at
           currentTime + t;

6  Event data message m received from neighbor s
7  if m is a duplicate then
8      if rebroadcast_timer for m is scheduled then
9          cancel rebroadcast_timer for m;

10 Event scheduled rebroadcast_timer for m expires
11 Rebroadcast m;
```

When a vehicle receives a beacon b from a neighbor s, it verifies whether there is a data message that has not been acknowledge by s in b (lines 1–3). For that, the vehicle looks into its list of received messages and compares their IDs with the IDs contained in b. If the vehicle finds any message m that has not been acknowledged, then it calculates a waiting delay t to rebroadcast m (Line 4). Once again, such delay will depend on the social metric employed, as described in Section 3.3. After calculating the waiting delay, the vehicle schedules to rebroadcast m with delay t (Line 5). As in the broadcast suppression algorithm, while the vehicle is scheduled to rebroadcast m, if it receives a duplicate, then it cancels the rebroadcast (lines 6–9), thus avoiding a possible redundant retransmission. However, when the waiting delay expires and the vehicle has not received any duplicate, then it rebroadcasts m (lines 10–11).

By using these two algorithms in conjunction, our proposed solution is able to tackle both the broadcast storm and the intermittently connected network problems. Moreover, it is worth noticing that a vehicle does not need to be aware of the current road traffic condition, i.e., whether the network is dense or sparse. In either case, the vehicle always tries to avoid redundant retransmissions and increase the message delivery capability to intended recipients.

3.3 Socially Inspired Dissemination

As outlined in the previous algorithms, calculating the waiting delay to rebroadcast a data message is the key step. Therefore, in this section, we show how the clustering coefficient and node degree can be used to turn our data dissemination solution into a socially-aware proposal. Initially, we show how to estimate the clustering coefficient using only one-hop neighbor information, and how to use it in the waiting delay computation. We then turn our attention to the node degree, which can be easily obtained through beacons. Finally, we also show how to calculate the waiting delay using a combination of both metrics.

3.3.1 Clustering Coefficient

The clustering coefficient for a vehicle v is the number of connections between neighbors of v divided by the total number of possible connections between neighbors of v. Therefore, to accurately calculate the clustering coefficient for vehicle v, it is necessary to know the two-hop neighborhood knowledge of v. Given that VANETs are extremely dynamic networks and obtaining such knowledge can be cumbersome, here we use position information to estimate the clustering coefficient, in particular, to determine whether two neighbors of a vehicle are connected or not. As already stated, each vehicle knows the position of each neighbor due to received beacons. Therefore, to verify whether two neighbors are connected or not, vehicle v must only check whether the distance between these two neighbors is below the estimated communication range. Thereafter, v is able to calculate its estimated clustering coefficient.

In possession of its own estimated clustering coefficient, a vehicle v is able to calculate its waiting delay to rebroadcast. According to an analysis of the estimated clustering coefficient with respect to the vehicle density (see Figure 2), for lower densities, the clustering coefficient is also low, but the variability is high. On the other hand, when the density is high, also is the value for the estimated clustering coefficient, but the variability is low. For our purposes, the greater the variability, the better. Otherwise, we risk assigning the same or similar waiting delay to all vehicles. Therefore, for this first proposal, we give a higher priority to rebroadcast for vehicles that have a low estimated clustering coefficient. In other words, the lower the estimated clustering coefficient, the lower the waiting delay. We calculate the waiting delay by the equation: $t_{cc} = T_{max} \times estimatedCC$, where the value for *estimatedCC* ranges in the interval [0, 1].

3.3.2 Node Degree

When we look into the analysis of the node degree (see Figure 2), we can see that, when the vehicle density is low, the degree and its variability is also low. However, when the density increases, both the degree and its variability increase. Therefore, in the proposal based on the node degree, we use an opposite approach. That is, the higher the degree of a vehicle at a given neighborhood, the higher its priority to rebroadcast the message, i.e., the lower the waiting delay. Each vehicle will know its max neighbor degree due to the *degree* information in the received beacons. Equation 1 shows how the waiting delay is calculated using this approach. Here, *degree* is the degree of the vehicle that is calculating the waiting delay and *maxDegree* is the maximum between *degree* and the highest degree among all neighbors of the vehicle.

$$t_{degree} = T_{max} \times \left(1 - \left(\frac{degree}{maxDegree}\right)\right) \quad (1)$$

3.3.3 Joint Solution

Here, we also propose a joint solution, i.e., one that uses both the estimated clustering coefficient and the node degree. The idea

is that, assuming that a single metric may not be adequate for all traffic density scenarios, a combination of the two may produce better results. The waiting delay can be calculated using this joint approach, defined by the equation: $t = \alpha t_{cc} + \beta t_{degree}$. As can be observed, each metric contributes to a fraction of the total waiting delay, which is controlled by the factors α and β. In this work, to balance the equation delay, we assume that $\alpha = \beta = 0.5$.

4. PERFORMANCE EVALUATION

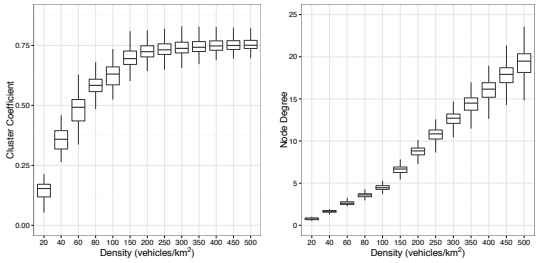
To evaluate the performance of our proposed approaches, we performed a series of simulations using the OMNeT++ 4.2.2. simulator [10]. We compare them to two well-known protocols - UV-CAST [11] and ABSM [6]. Moreover, in the presented results, the socially inspired protocols are identified as CC (clustering coefficient), Degree (node degree) and CC-Degree (joint solution). In the following sections, we describe the performance evaluation in detail. In particular, Section 4.1 shows the scenarios and the default parameters used in our analysis. Finally, simulation results and discussion are presented in Section 4.2.

4.1 Simulation Setup

Manhattan scenario: this is a scenario with ten evenly-spaced double-lane streets in an area of 1 km^2. Also, we consider signal attenuation effects caused by buildings. For that, we assume that each block has an 80m $\times$ 80m obstacle, which represents high-rise buildings. In order to quantify the traffic evolution in this scenario, we vary the vehicle density from 20 vehicles/km^2 to 500 vehicles/km^2. The road traffic simulation is performed by the Simulator of Urban MObility (SUMO 0.17.0) [1]. Moreover, we positioned the source vehicle at the center of the grid and it generates 100 messages of 2048 bytes to be disseminated to the whole network. The data rate is set to 1.5 Mbit/s.

To better understand the Manhattan grid scenario, Figure 2 shows the estimated cluster coefficient and the node degree for the considered vehicle densities. In particular, Figure 2-(a), we show the estimated cluster coefficient and its evolution. As we can see, the value for the estimated cluster coefficient under low densities is small, almost 40 %. Moreover, it has a higher variability. It happens because, for lower densities, there are few vehicles in transit. With the growth of the density, the estimated cluster coefficient increases. This is due to the fact that, under higher densities, the encounter probability is also higher, and the network will be more connected. Therefore, starting at 200 vehicles/km^2, the value for the estimated clustering coefficient has a constant behavior of about 75 %. This can be explained by the fact that even if the density of the network increases, connections among vehicles are constrained by physical restrictions, such as road shapes and obstacles. The Figure 2-(b) presents the node degree evolution. It is possible to observe how the node degree evolves over the density variation. As expected, with the increase of the density, the node degree also increases. For instance, at 100 vehicles/km^2, the average node degree is 5, representing that, on average, a vehicle has 5 neighbors.

Finally, to improve the quality of the following results and make them more realistic, we rely on the Veins 2.1 [8] network framework. It implements the standard IEEE 802.11p protocol stack for vehicle communication and an obstacle model for signal attenuation. Moreover, we set the bit rate at the MAC layer to 18 Mbit/s and the transmission power to 0.98 mW. With these parameters and a two-ray ground propagation model, it is possible to reach a communication range of 200 m. Beacons are sent every 1 s. For all scenarios, we simulate r replications in order to compute the confidence interval of 95%. According to the definitions presented in [5], the metrics evaluated are: *Delivery ratio*, *Total Messages Transmitted*, *Collisions* and *Delay*. The focus is to verify the coverage of



(a) Estimated cluster coefficient (b) Node degree evolution

Figure 2: Manhattan metrics evolution.

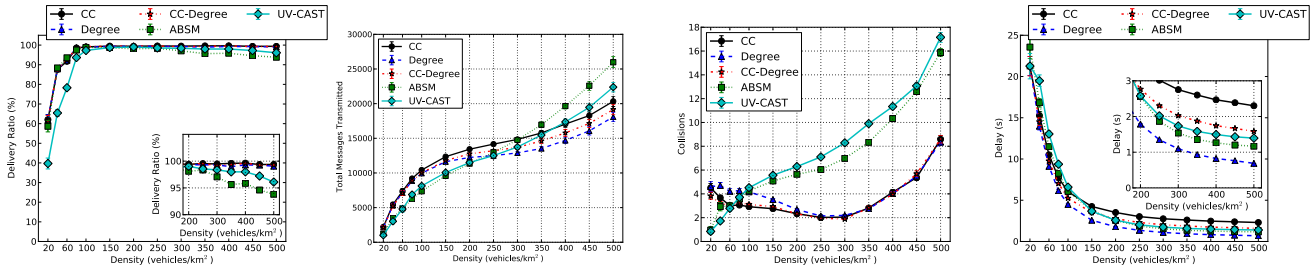
the protocol, the overhead induced by data messages and the latency for different network density conditions.

4.2 Results

Figure 3 shows the results for the Manhattan grid scenario. As we can note, overall, our socially inspired approaches present a better performance. When considering the delivery ratio (Figure 3-(a)), for lower densities, we can observe that CC, Degree, CC-Degree and ABSM deliver data messages to the same amount of vehicles. As the density increases, so does the delivery results for all protocols. However, for very high densities, the performance of ABSM and UV-CAST starts to deteriorate, while our proposals guarantee 100% delivery ratio. In summary, this result shows that the considered social metrics leads to the same delivery capability.

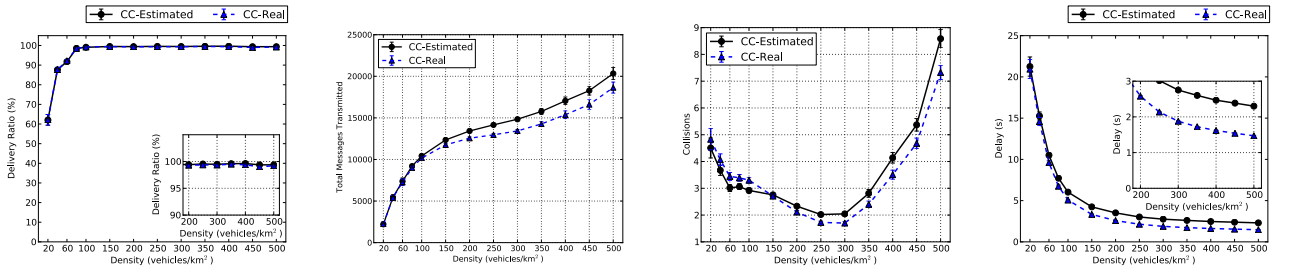
Figure 3-(b) shows the number of data messages transmitted. For lower densities, our proposals transmit more data messages when compared to ABSM and UV-CAST. As shown in the previous result, given that CC, Degree, CC-Degree and ABSM have the same delivery results for such lower densities, we can conclude that our proposals are not able to avoid redundant retransmissions when the network is sparse. Notice that, the broadcast storm problem is not much an issue in sparse networks. As the density increases, our solutions incur the lowest number of data messages transmitted. Among the three, the degree presents the best results, while the CC the worse. Recall from the results shown in Figure 2 that, at higher densities the variability for the degree is higher when compared to the one presented by the clustering coefficient. As already stated, the greater the variability, the greater the range of possible waiting delays, which leads to a better broadcast suppression approach. In a similar result, Figure 3-(c) shows the number of collisions for all protocols. Essentially, the behavior is almost the same for the number of messages transmitted. Our approaches perform better at higher densities. It is worth noticing that, at lower densities, among our solutions, the Degree leads to the highest number of collisions, while the CC leads to the lowest. This fact can also be explained by the variability results shown in Figure 2.

Figure 3-(d) shows the delay for all approaches. As expected, for lower densities, the delay for all protocols is very high due to the store-carry-forward performed by all protocols, i.e., vehicles need to store and carry messages around in order to deliver them. As the density increases, the delay for all protocols decreases. In particular, Degree has the lowest delay, while CC has the highest. According to the results shown in Figure 2, for higher densities, the clustering coefficient is also high. Therefore, the waiting delays chosen by vehicle will also be high, thus explaining the higher average delay. In the case of the Degree, for higher densities, the node degree is also high. However, contrary to the clustering coefficient, nodes with a high degree have a lower waiting delay, which explains the average delay to deliver data messages to intended recipients.



(a) Delivery Ratio (b) Messages Transmitted (c) Collisions (d) Delay

Figure 3: Simulation results for the Manhattan street scenarios.



(a) Delivery Ratio (b) Messages Transmitted (c) Collisions (d) Delay

Figure 4: Comparison for performance evaluation between cluster coefficient estimated and real for the Manhattan scenario.

Finally, aiming to analyze the difference between the estimated cluster coefficient and the real cluster coefficient, we present the results for these metrics in Figure 4 under the Manhattan grid scenario. Recall that, the estimated clustering coefficient is computed by considering the distance between vehicles and the estimated communication range. Therefore, signal attenuation caused by buildings has a direct impact on it. Conversely, the real clustering coefficient is calculated using the two-hop neighborhood knowledge of vehicles. As can be observed, overall, using the real clustering coefficient results on a better performance. However, the difference to the results of the estimated clustering coefficient are not significant, especially when we consider the extra cost to compute the real clustering coefficient.

5. CONCLUSION AND FUTURE WORK

This paper presented a new approach to perform data dissemination in vehicular ad hoc networks that take into account the social aspects of the network and the traffic evolution. Aiming to solve the broadcast storm and the intermittently connected network problems, our approach uses the information about the number of neighbors (node degree) and how these neighbors are connected (clustering coefficient) as a criterion to define which vehicles rebroadcast data messages during the dissemination process. We evaluated the performance of our approach, and we noted that in sparse scenarios our approach presents a performance a little worse when compared with related protocols. However, in high density scenarios, our approach presented substantial performance gains, especially regarding the delivery capability and the incurred overhead. As future work, we intend to investigate how other social metrics may be used to develop better data dissemination solutions and more efficient communication protocols.

6. REFERENCES

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

[2] F. Cunha, A. Carneiro Viana, R. A. F. Mini, and A. A.F. Loureiro. Is it possible to find social properties in vehicular networks? In *IEEE Symposium on Comp.and Com.(ISCC '14)*, 2014.

[3] M. Fiore and J. Härri. The networking shape of vehicular mobility. In *ACM Int. Symposium on Mobile Ad Hoc Net. and Computing (MobiHoc '08)*, pages 261–272, 2008.

[4] D. Naboulsi and M. Fiore. On the Instantaneous Topology of a Large-scale Urban Vehicular Network: The Cologne Case. In *ACM Int.l Symposium on Mobile Ad Hoc Net. and Computing (MobiHoc '13)*, pages 167–176, 2013.

[5] S. Panichpapiboon and W. Pattara-Atikom. A review of information dissemination protocols for vehicular ad hoc networks. *Com. Surveys Tutorials, IEEE*, 14(3):784–798, Third 2012.

[6] F. Ros, P. Ruiz, and I. Stojmenovic. Acknowledgment-Based Broadcast Protocol for Reliable and Efficient Data Dissemination in Vehicular Ad Hoc Networks. *IEEE Transactions on Mobile Computing*, 11(1):33–46, 2012.

[7] R. S. Schwartz, R. R. R. Barbosa, N. Meratnia, G. Heijenk, and H. Scholten. A directional data dissemination protocol for vehicular environments. *Computer Com.*, 34(17):2057–2071, 2011.

[8] C. Sommer, R. German, and F. Dressler. Bidirectionally Coupled Network and Road Traffic Simulation for Improved IVC Analysis. *IEEE Transactions on Mobile Comp.*, 10(1):3–15, 2011.

[9] O. K. Tonguz, N. Wisitpongphan, and F. Bai. DV-CAST: a distributed vehicular broadcast protocol for vehicular ad hoc networks. *IEEE Wireless Com.*, 17(2):47–56, 2010.

[10] A. Varga and 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)*, pages 1–10, 2008.

[11] W. Viriyasitavat, O. Tonguz, and F. Bai. UV-CAST: an urban vehicular broadcast protocol. *IEEE Com. Mag.*, 49(11):116–124, 2011.