



HAL
open science

Adaptive HELLO Protocol for Vehicular Networks

Nathalie Mitton, Yasir Saleem, Valeria Loscri, Christophe Bureau

► **To cite this version:**

Nathalie Mitton, Yasir Saleem, Valeria Loscri, Christophe Bureau. Adaptive HELLO Protocol for Vehicular Networks. ITU Journal on Future and Evolving Technologies, 2024. hal-04459157

HAL Id: hal-04459157

<https://inria.hal.science/hal-04459157v1>

Submitted on 15 Feb 2024

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.



Distributed under a Creative Commons Attribution 4.0 International License

Adaptive HELLO Protocol for Vehicular Networks

Nathalie Mitton
Inria, France
nathalie.mitton@inria.fr

Yasir Saleem
Aberystwyth University, UK
yasir.saleem@aber.ac.uk

Valeria Loscri
Inria, France
valeria.loscri@inria.fr

Christophe Bureau
#dot:dot, France
xtof@dotdot.fr

Abstract—In vehicular-to-vehicular networks, the update of car Firmwares Over-The-Air (FOTA) is becoming a challenging issue. Indeed, it mainly relies on topology discovery of neighbouring nodes. Topology discovery in mobile wireless networks is usually done by using HELLO messages. Due to mobility, topology changes occur frequently and must be quickly discovered to avoid routing failures. Since the optimal HELLO frequency depends on parameters that are subject to changes (e.g., speed of nodes, density of nodes), it must be dynamically adjusted to obtain the best trade-off between the network load and the freshness of routing tables. Existing solutions mainly focus on general wireless networks assuming random mobility and constant node density and average speed. Such assumptions do not hold in the case of vehicular networks in which vehicles follow specific trajectory patterns (the roads) and density and speed evolve as a function of time (rush hour vs non-rush hour) and area (urban, rural, highway). In this paper, we first draw the specific features of a vehicular network at different times and spaces by analysing real datasets. This study then allows us to adapt a dynamic neighbour discovery protocol to the specific case of vehicular-to-vehicular communications to propose VANP (Vehicular Adaptive Neighbour Discovery Protocol). VANP is a fully distributed and software-based solution that sends beacons at an optimal frequency without knowing it a priori. The objective is to reduce the frequency at which HELLO messages are sent to save bandwidth and energy while still preserving the quality of the neighbour discovery. Through extensive simulations run on real datasets, we show that the optimal HELLO frequency can be reached by maintaining a constant optimal turnover, independent from the speed of the nodes and by aiming at this turnover, nodes automatically use the optimal HELLO frequency. Results show that VANP allows the discovery of relevant neighbours by missing at most two neighbours over all scenarios and reducing the number of HELLO messages up to twice, hence saving bandwidth and energy.

I. Introduction

The update of car Firmwares Over-The-Air (FOTA) is a growing trend that transforms software related recalls into mere updates, thus considerably reducing the cost of recalls for car manufacturers, while increasing driver safety in quasi real time. However, with the steep increase in the number of processors per vehicle, combined with the increase in the number of connected and electric vehicles, the number and data volume of updates of Firmwares Over-The-Air (FOTA) is expected to increase exponentially. Multiplied by the increasing cost of mobile data subscriptions, this leads to the paradox that by 2030 or even before that, it may be more costly to update vehicles than to move them if FOTA is only relying on mobile network communication (be it LTE/4G or 5G)

[1]. In addition, a significant portion (15 to 20%) of the reduction in CO2 footprint brought by (connected) electric vehicles is ruined by the carbon impact of mobile networks and cloud infrastructure [2]. There is thus a need to find a credible alternative to mobile networks for FOTA's, from the ones that occur on the parking of the factory to the ones taking place when cars are already on the road. One appealing solution resides in the use of through-car mesh networks: car crowds are “quasi-networks”, only missing an ubiquitous mesh organisation with a radio network providing sufficient throughput. Such mesh networks are collaborative in nature, and much more open to creative business models without subscriptions, hence with the potential to bypass all or part of the cost of data subscription that presently makes FOTA appear like a dead-end. In addition, a mesh network being intrinsically more efficient than a global LTE network, one can also expect a significant reduction in the carbon footprint of FOTA using mesh networks rather than mobile-based networks. In such a mesh, vehicular-to-vehicular (V2V) communications are dynamically created when the vehicles are close enough to one another, and no pre-established infrastructure is required. Due to the path loss of radio communications, only close vehicles may directly communicate to each other. Long-distance communications require multi-hop routing, where packets are forwarded by multiple intermediate nodes. Many efficient localised routing and data forwarding protocols rely on a local neighbourhood table. This table is created by each node u and contains the list of nodes with which u can directly communicate (its neighbours). It is generally maintained by letting nodes broadcast HELLO messages to their neighbours.

The efficiency of table-driven protocols (e.g., routing, clustering, activity scheduling) obviously relies on the accuracy of these tables. In a mobile environment, neighbourhood tables are subject to change, and in this case, the accuracy depends on the frequency of HELLO messages. If this frequency is too low, nodes may not be detected by their neighbours, leading deprecated neighbourhood tables, and protocol failures are likely to occur. On the other hand, if the frequency is too high, neighbourhood tables are up to date, but then energy and bandwidth are wasted to the detriment of data traffic.

Yet, determining the correct frequency is not obvious. The optimal value, providing a good trade-off between

these behaviours, actually depends on dynamic characteristics, such as the speed of nodes, and should thus be dynamically updated. By optimal frequency, we mean the minimum frequency at which HELLO messages can be sent to ensure that nodes correctly detect each other. The most straightforward solution is to suppose that nodes know their speed, but this assumption requires dedicated hardware (e.g., GPS) that may not be available.

In this paper, we first analyse the behaviour of a vehicular network in different environments (urban, highway) at different times (rush hour, non-rush hour) by running extensive real dataset based experimentation. Thanks to them, we could observe among others that each node discovers a non negligible part of neighbours with which it does not stay in range long enough to initiate some data exchanges. In a vehicular network application, we can thus distinguish the “relevant neighbours” from the set of all neighbours. We define the “relevant neighbours” as the neighbours with which we remain in range of each other long enough to exchange data. By adapting and decreasing the HELLO frequency, we consequently may miss neighbours. What we claim in this study is that this is not mandatory to discover all of them, it is even desirable since we mainly need to discover “relevant neighbours”.

These important observations have driven our study and the adaptation brought to TAP (Turnover-based Adaptive HELLO Protocol), an existing adaptive neighbour discovery protocol designed for general wireless networks [3]. Our goal is to set proper parameters in our HELLO frequency adaptation algorithm in order to reduce HELLO frequency when possible without jeopardising the efficient discovery of “relevant” neighbours. As a result, we propose VANP, a vehicular adaptive neighbour discovery protocol. The key features of VANP are as follows :

- It is localised, and thus topology changes have very limited impact.
- It is fully software-based, and does not require dedicated hardware (such as a GPS).
- It is independent from the communication stack (e.g., routing, MAC).
- It is light on CPU usage as no on-node heavy computation is required.
- It specifically suits vehicular networks and adapt to all road environments.

The contributions of this paper can be summed up as follows:

- a road traffic analysis based on real datasets that shows that it is not useful to discover all neighbours every time,
- the VANP protocol, an adaptation of the TAP protocol to the vehicular context,
- a complete simulation-based evaluation relying on real datasets showing that VANP allows the discovery of relevant neighbours by missing at most two neighbours over all scenarios and reducing the number of

HELLO messages up to twice hence saving bandwidth and energy.

The outline of this paper is as follows. After discussing related works in Section II, Section III presents the motivation of this work. Section IV provides a background of TAP. In Section V, we describe our protocol and discuss a few implementation issues, mainly about the accurate estimation of the turnover. Section VI provide some experimental results run over real datasets and presents the performance of VANP. We finally conclude and discuss open issues in Section VII.

II. Related works

Hello protocols and neighbours discovery have been the focus of several existing studies since networks exist. In this related work review, we mainly focus on neighbour discovery for vehicular networks [4], [5], [6].

The authors of [7] propose REMR, a position-based routing scheme for emergency messages which adapts the HELLO interval based on the neighbourhood density to minimise HELLO messages congestion. The HELLO interval is initialised to 0.03s. Before sending a HELLO message, a vehicle checks whether the number of its neighbours is greater than the average number of neighbours in the network (considered as a threshold value). If so, the network is considered as high-density and the vehicle adapts the new HELLO interval by adding the ratio of the number of vehicles at current and previous time steps to the HELLO interval. Otherwise, it sets the HELLO interval to 0.03s. The limitation of this work is that there is no means to reduce the HELLO interval except resetting it to 0.03s, which is not always pertinent and assumptions do not always hold.

Li [8] proposed the adaptation of HELLO interval using cooperative scheme for vehicular networks based on the acknowledgements of HELLO messages. The acknowledgements are used to identify HELLO messages loss. When a vehicle detects HELLO messages loss, it increases its HELLO frequency to retain the neighbourhood information. The limitation of this work is that vehicles are required to know their neighbours for receiving acknowledgements and the medium is quickly saturated by multiple acknowledgements.

In [9], rather than adapting the HELLO interval, the authors investigate adapting the transmission power of HELLO messages based on the prediction error of vehicle position. The vehicles increase their transmission power of HELLO messages when prediction errors are high, while reduce their transmission power on lower prediction errors. The simulations results showed reduced channel busy time and higher packet transmission rate compared to fixed transmission power of HELLO messages but they pursue a different goal.

Lyu et al. [10] worked on an adaptation of HELLO interval to avoid rear-end car crash by defining a danger coefficient which identifies the danger threat of each

vehicle to be involved in a rear-end collision. Based on the danger coefficient, the authors proposed a distributed HELLO messages congestion control strategy to adapt the HELLO interval. Each vehicle calculates its danger coefficient and shares with neighbouring vehicles through HELLO messages. The vehicles having higher danger coefficient use lower HELLO interval to allow taking immediate actions by rear vehicles in case of sudden braking to avoid collisions.

Li et al. [11] used mobility prediction to adapt the HELLO interval and to reduce channel occupancy due to the frequent broadcasting of HELLO messages. The authors claim that vehicles can predict the position of neighbouring vehicles. The vehicles use Kalman Filter to predict the future possible moving states and use kinematics law to predict the mobility of their neighbouring vehicles. The HELLO interval is adapted by comparing the prediction error to a deviation threshold. If prediction error is higher than the deviation threshold, the HELLO messages are broadcast, otherwise the HELLO interval is reduced.

Nguyen and Jeong [12] investigated adaptive HELLO broadcast using mobility prediction for guaranteed tracking of degree (GTK) K . The allowed tracking inaccuracy that does not affect the reliability of vehicle collision avoidance is called tracking threshold. Given a tracking threshold, GTK K is defined as the guarantee that a vehicle that receives at least one of the K HELLO messages from a sending vehicle, should be able to estimate the position of sending vehicle within the accuracy of tracking threshold. The main objective of authors' scheme is to postpone the broadcasting of HELLO message as long as the tracking inaccuracy is less than the tracking threshold. This saves the frequent broadcasting of HELLO messages.

In [13], the authors investigate rear-end collision and proposed BASE for the adaptation of HELLO interval to save the radio resources without jeopardising the safety requirements. BASE adapts the HELLO interval of each vehicle by guaranteeing that a minimum safety distance should always be maintained between vehicles. The minimum safety distance is defined as the required distance to avoid rear-end collision caused by sudden brake of vehicle in front. BASE is mainly focused on normal traffic conditions and therefore, does not consider complicated conditions, such as crossroads and overtaking.

In [14], the authors worked on congestion control by investigating the adaptation of HELLO interval and awareness control using non-cooperative game theory and proposed NORAC. NORAC assigns a HELLO interval to each vehicle fulfilling its needs by guaranteeing fairness among vehicles having similar needs. NORAC is distributed and non-cooperative. It guarantees fairness using the fair concept of Nash Equilibrium.

Unlike these works, our approach does not require heavy-computing or predictions. It is localised and fully adaptive.

III. Motivation : vehicular networks study

In order to characterise key features of a vehicular network, we have run a simulation based study by analysing real traffic datasets. We have used the SUMO (Simulation of Urban MObility) [15] road traffic simulator integrated with OMNeT++ [16], an event-driven network simulator to simulate the vehicular wireless communications and Veins [17], an open-source framework that provides the functionality of vehicular network.

Various real-world road traffic scenarios of different cities have been made available by researchers¹ for SUMO [15] that can be imported to various network simulators (e.g., OMNeT++, NS-3). We conducted this analysis over different cities. In this paper, we only present the results for the city of Dublin over different scenarios but results are similar for different cities.

A. Scenarios description

In this paper, we mainly focus on the scenarios of rush and non-rush hours of Dublin City Centre (referred as Ireland urban scenario hereafter) and Ireland National Road (referred as Ireland national highway hereafter). The rush hour (saturated traffic) is when the level of traffic is high and it starts getting congested. The non-rush hour (free flow traffic) is early morning traffic when the level of traffic is low.

1) Ireland Urban Scenario: Ireland urban scenario is comprised of 5km×3.5km Dublin city centre having 435 signalised intersections. Dublin SCATS dataset², a traffic light control system, was used to generate the traffic by counting the vehicles using SCATS sensors every six minutes at 480 locations in Dublin city. It provides traffic for 24 hours. The traffic between 8am–9am and 4am–5am are the rush and non-rush hours traffic, respectively [18].

2) Ireland National Highway Scenario: Ireland National Highway is N7, a 3-lane two-way road of 17.1 km long having 11 on- and off-ramps each way. The loop sensors from open dataset were used to generate the traffic and the Transport Infrastructure Ireland³ published the data that includes 5-minute aggregated data for each lane and direction for several months. The traffic is averaged to create a workday traffic by excluding weekends and bank holidays. It provides traffic for 24 hours. The traffic between 5am–6am and 4am–5am are the rush and non-rush hours traffic, respectively [18].

B. On the pertinence of relevant neighbours

As previously discussed, we claim that it is not useful to detect all neighbours at everytime but only relevant neighbours, i.e. the ones that allow a contact long enough for a data transmission. To illustrate this, Figure 1 shows for each environment (urban rush hour, non-rush hour, highway rush hour, non-rush hour) the distribution of

¹<https://sumo.dlr.de/docs/Data/Scenarios.html>

²<https://data.gov.ie/dataset/traffic-volumes>

³<https://traffidata.tii.ie/publicmultinodemap.asp>

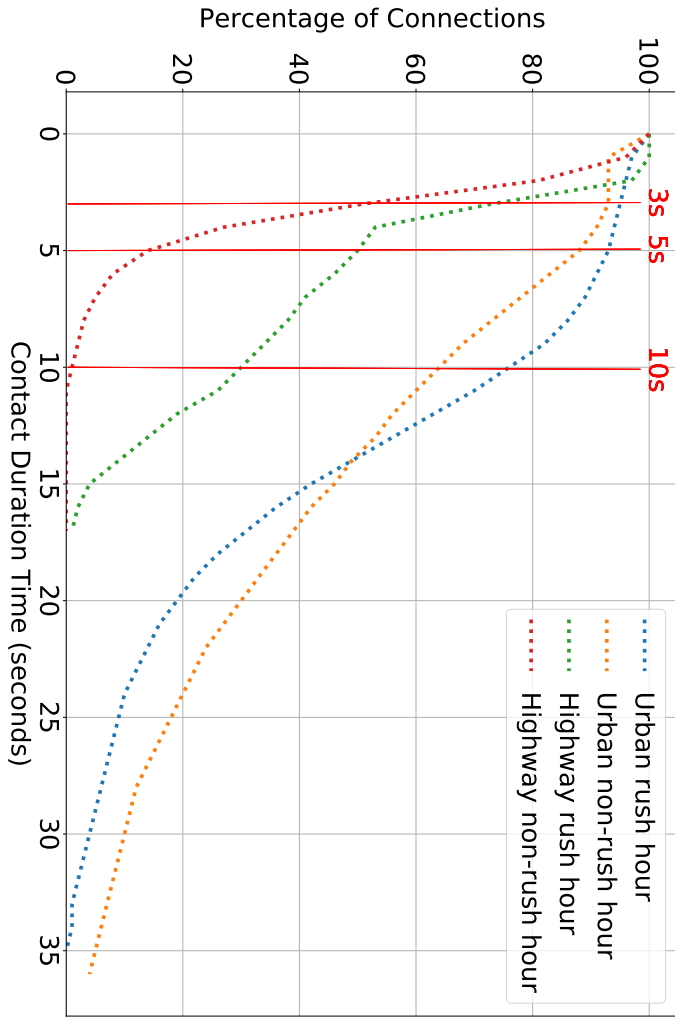


Fig. 1: Distribution of contact time.

the average contact duration of connections between two vehicular neighbours. We can see for instance on this figure that in an urban environment at rush hour time (orange curve) that 50% of connections last for less than 12s and that only 20% of connections last more than 20s. Applications in vehicular networks generally require the exchange of data and thus a stable contact during a minimum time. Vertical lines on figure mark the limits for connections that last respectively more than 3, 5 or 10s. If we define the relevant neighbours as the neighbours that remain in contact for at least x seconds, we can see that in urban environment in rush hour, only 93% neighbours are relevant for $x = 3$, 90% for $x = 5$ and that this amount drops to 76% for $x = 10$ s.

This demonstrates that there is no use to design a neighbour discovery protocol that has a constant complete view of neighbourhood since part of the nodes will not be usable.

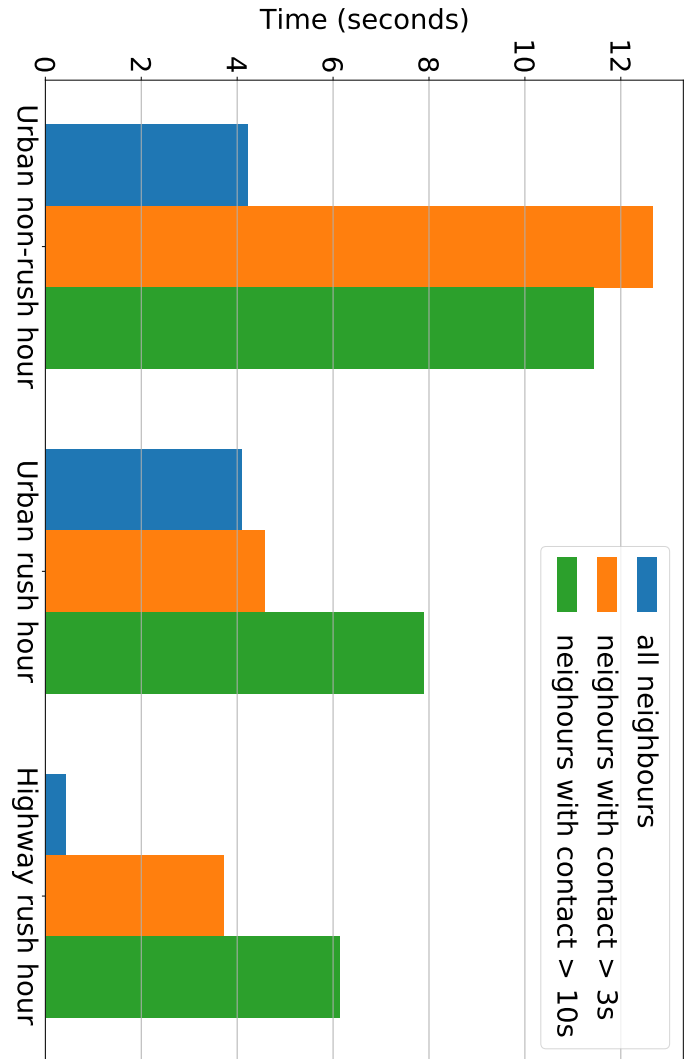


Fig. 2: Average time before meeting a new relevant neighbour

C. On the HELLO frequency adaptation

Figure 2 shows for each environment the average time before meeting a new relevant neighbour. As expected, it highlights that in average, a node needs much more time before meeting a relevant neighbour than any single neighbour. As such, this means that the delay between two HELLO messages can be increased. This motivates the need to provide an adaptive Hello protocol specific to the vehicular context.

Table I sums up the notations used in this paper.

IV. Background: the TAP protocol

In this section, we provide a summary of the TAP protocol [3] that lays the basis of VANP. Let us assume that nodes send HELLO messages at the frequency f_{HELLO} (a HELLO message is sent every $d_{\text{HELLO}} = \frac{1}{f_{\text{HELLO}}}$). Whenever a node receives a HELLO message, it updates its neighbourhood table, thus generating some turnover

TABLE I: Notations.

Notation	Description
r	Current observed turnover
r_{opt}	Optimal turnover
f_{HELLO}	Hello frequency ($f_{\text{HELLO}} = \frac{1}{d_{\text{HELLO}}}$)
d_{HELLO}	Delay between two Hello messages ($d_{\text{HELLO}} = \frac{1}{f_{\text{HELLO}}}$)
f_{opt}	Optimal Hello frequency
X	Size of history
N_t	Neighbourhood table at time t

r . Given a period of time Δt , we define the turnover $r_{\Delta t}$ to be the ratio between the number of new neighbours (i.e., nodes that were not already neighbours Δt units of time earlier) and the current total number of neighbours. Obviously, the turnover depends on both Δt and f_{HELLO} . Without loss of generality, let us set $\Delta t = 1/f_{\text{HELLO}} = d_{\text{HELLO}}$, that is, the turnover is updated each time a new HELLO message is sent. As is, a high frequency induces a low turnover (nodes frequently update their tables and thus discover few new neighbours each time), while a low frequency induces a high turnover. Because of that direct relationship between the two values, we can note that acting on one of them directly affects the other one. TAP relies on the idea that the optimal HELLO frequency f_{opt} generates an optimal turnover r_{opt} , and by keeping r close to r_{opt} , f_{HELLO} automatically tends toward f_{opt} . Nodes thus have to slightly adjust their frequency with regards to the observed turnover. Authors of [3] proved that while the optimal frequency depends on the speed of the node, the optimal turnover is fixed and does not depend on any such parameter. The optimal frequency indeed evolves accordingly to the speed (i.e., it increases when the speed increases, and decreases when the speed decreases), but eventually generates a constant turnover r which can be computed offline. TAP is also efficient in static networks.

V. Vehicular Adaptive Neighbour Discovery Protocol

As explained, VANP is inspired by TAP. However, TAP does not specify the way to adapt the frequency based on the observed turnover. Also, TAP has computed the optimal turnover r_{opt} in a general wireless network, assuming nodes are uniformly distributed in space and follow a random way point mobility model with fixed average speed, which is not the case in vehicular network. In this section, we describe VANP and in particular how it adapts the HELLO frequency. Then we compute the optimal turnover to be observed in our environment.

A. Implementation

1) Implementation: Obviously, accurately estimating the turnover is of prime importance. As we will show in the next section, the value of the optimal turnover may be quite low, and for low densities it can be difficult to effectively estimate the current turnover. For instance, if a node has only two neighbours at a given moment,

the turnover is limited to three distinct values (zero if none of the neighbours are new, one if only one of them is new, or two otherwise), which is not precise enough to accurately adjust the current HELLO frequency. A solution to this problem is to take account of the number of new neighbours that appeared at every X previous observations, X being the size of the history. In that case, the value may be obtained by enumerating new neighbours between two successive tables:

$$r = \frac{d_{\text{HELLO}}}{t_X - t_0} \sum_{i=1}^X \frac{\text{new}_{t_i - t_{i-1}}}{|N_{t_i}|}, \quad (1)$$

where $\text{new}_{t_i - t_{i-1}}$ is the number of new neighbours detected between times t_{i-1} and t_i , and N_{t_i} is the neighbourhood table at time t_i and $|N_{t_i}|$ its size. By doing this, it is possible to obtain a sufficient accuracy of the turnover estimation, as we will show in Section VI.

The adjustment of f_{HELLO} requires the use of an incremental update function based on the observed turnover r . A very simple series like the following one perfectly fits this purpose:

$$d_{\text{HELLO}} = \left\{ \begin{array}{ll} d_{\text{HELLO}} + \frac{d_{\text{HELLO}}}{\alpha} \times g(r) & \text{if } r \leq r_{\text{opt}}, \\ d_{\text{HELLO}} - \frac{d_{\text{HELLO}}}{\alpha} \times g(r) & \text{otherwise.} \end{array} \right\} \quad (2)$$

Here, α is a constant controlling the speed of the convergence, and $g(r)$ is a function of the observed turnover that returns how much d_{HELLO} should be changed. To design this function, we should first note that the more r and r_{opt} are different from each other, the more likely the current frequency f_{HELLO} and the optimal frequency f_{opt} are different from each other. Thus, the higher the difference between r and r_{opt} , the more quickly f_{HELLO} should “move” towards f_{opt} . We define $g(x)$ as follows:

$$g(r) = \left\{ \begin{array}{ll} \left(\frac{r - r_{\text{opt}}}{r_{\text{opt}}}\right)^2 & \text{if } r < 2 \times r_{\text{opt}}, \\ 1 & \text{otherwise.} \end{array} \right\} \quad (3)$$

When r and r_{opt} are only slightly different, then the normalised amplitude is close to 0, resulting in a slight variation of f_{HELLO} . Opposed to this, the higher the difference between r and r_{opt} , the higher the amplitude and the faster f_{HELLO} converges toward the optimal frequency f_{opt} . Any other function $g(r)$ such that $g(r_{\text{opt}}) = 0$ may of course be used. However, this one is very straightforward to implement, and a lookup table could actually be used to avoid overhead.

As a result, Algorithm 1 describes VANP .

B. Determining the optimal turnover r_{opt}

Now that the general design of our protocol has been presented, the principal issue that remains opened is: What is the optimal turnover r_{opt} ? Finding the correct value of f_{opt} is indeed not trivial, as it depends on a lot of parameters. To deduce it, we rely on our real dataset analysis.

Algorithm 1 Adaptive Hello Protocol.

```

1: Initialise  $r_{opt}$ ,  $d_{HELLO}$ ,  $X$ ;
2:  $i = 1$ ;
3: while true do
4:    $new = \frac{1}{i} \sum_{j=0}^{i-1} |N_{t_{j+1}} \setminus N_{t_j}|$ 
5:    $r = new \times \frac{d_{HELLO}}{t_i - t_0}$ 
6:   Compute  $g(r)$  using Eq. 3
7:   Compute  $d_{HELLO}$  using Eq. 2
8:   if  $i < X$  then  $i++$ 
9:   end if
10:  Schedule Hello message after  $d_{HELLO}$  period;
11: end while

```

TABLE II: r_{opt} values.

	Urban		Highway	
	Rush hour	Non-rush hour	Rush hour	Non-rush hour
> 3s	0.85	0.85	0.5	0.2
> 10s	0.6	0.6	0.4	0.2

Figure 3 shows the proportion of relevant neighbours over time for each scenario. In this figure, we considered as relevant neighbours the ones that remain in contact for more than 3 and 10s respectively. Globally we can observe that whatever the scenario and thus the density and speed of nodes, this proportion remains constant over time and is the same for different times in the urban scenario (about 85% of neighbours in contact for more than 3s and 60% in contact for more than 10s). Third observation is that in highway scenarios where there are less intersections, the gap between the two categories of relevant neighbours is small (in rush hours, 50% of neighbours in contact for more than 3s against 40% for more than 10s and in non-rush hours, for all kind of relevant neighbours 20% of neighbours only). Obviously, when the vehicles can drive fast like in non-rush hour and this proportion allows us to determine the optimal turnover r_{opt} value to be used in the Hello adaptation process.

Based on these values, we retain the r_{opt} values depicted in Table II.

VI. Experimental Results

In this section, we analyse first the behaviour of VANP and then its performances. We use the same set up and simulation environment as detailed in Section III.

A. Protocol behaviour

Fig. 4 presents the behaviour of the delay between two HELLO messages for each non-rush hour scenario. We can observe that the delay between two HELLO messages dynamically change over time, as expected.

Figure 5 shows the resulting average d_{HELLO} in every non-rush hour scenario at the end of the simulation. One can observe that for each case, $d_{HELLO} > 0.5s$ the usual d_{HELLO} applied in communication protocols. This demonstrates that VANP sends less HELLO messages than classical protocols in urban environment and thus

save energy and bandwidth. At the contrary, more packets are sent in high speed environment (e.g. highway in non-rush hour) in order not to miss important neighbours. We can also observe that as expected, more packets are sent to discover neighbours that remain in contact for more than 3s than for neighbours that remain in contact for more than 10s (delay is smaller) since more nodes have to be spotted.

We can also note that the size of history X is not anodyne since using $X = 10$ instead of $X = 5$ allows even more saving and the saving is even more important in high speed scenarios as in highway scenario. This is because a greater history size allows to smooth rapid changes.

B. Protocol performances

Saving energy and bandwidth is paramount but not at the cost of neighbour tables accuracy. Therefore, Fig. 6 presents the amount of false positive relevant neighbours (Relevant neighbours in the neighbourhood table but not in communication range) in different scenarios for non-rush hour.

Whatever the scenarios, we can observe that VANP is very accurate since it never badly tags more than one node at a single time in the worst case and in average 0.2 nodes (so no bad tagging in most of cases).

Similarly, Fig. 7 presents the amount of missed relevant neighbours detections (Relevant neighbours in communication range but not in the neighbourhood table) in different scenarios for non-rush hour.

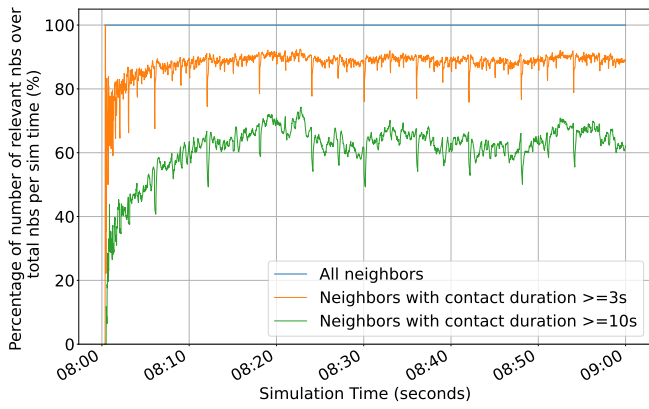
As previously, we can observe the high performance of VANP since it never miss more than two node at a single time in the worst case and in average 0.5 nodes (so no missed detection in most of cases).

VII. Conclusion

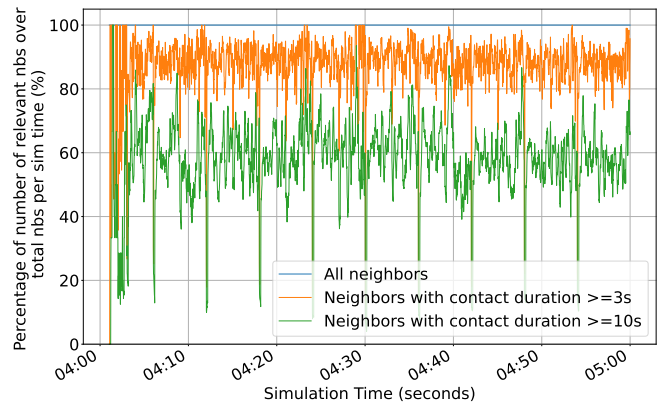
We have presented an adaptive HELLO protocol for vehicular networks that estimates the neighbourhood turnover and adjusts the HELLO frequency consequently. Besides the fact that our protocol is very straightforward to implement, it is especially well-tailored to standard mobile wireless networks since it does not rely on any specific hardware to aim at an optimal HELLO frequency. We have shown under realistic scenarios using real datasets that VANP is very accurate and allows substantial energy and bandwidth saving. Future work will focus on the automatic detection of the scenario (e.g. urban vs highway, rush hour vs non-rush hours) in order to dynamically adapt the optimal parameter setting to the network density and traffic model.

References

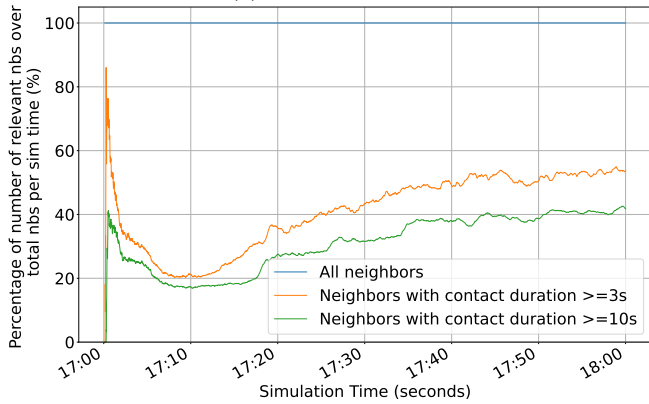
- [1] #dot:dot, “Car fota based on mobile networks: A dead end?” <https://beeyond.fr/s/4145071855>, 2023.
- [2] —, “Estimating carbon emissions in mobile networks,” <https://beeyond.fr/s/4145074127>, 2023.



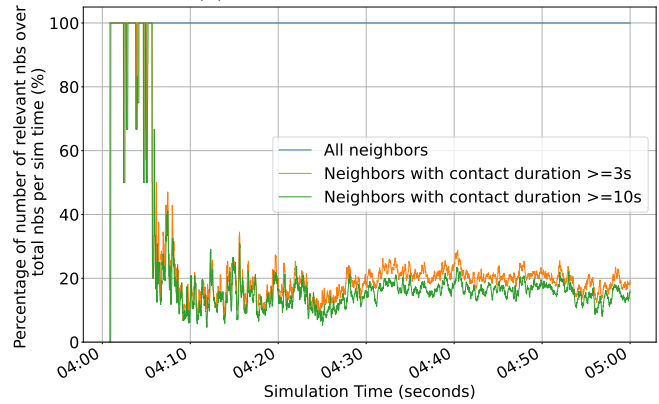
(a) Urban Rush hour.



(b) Urban non-rush scenario.

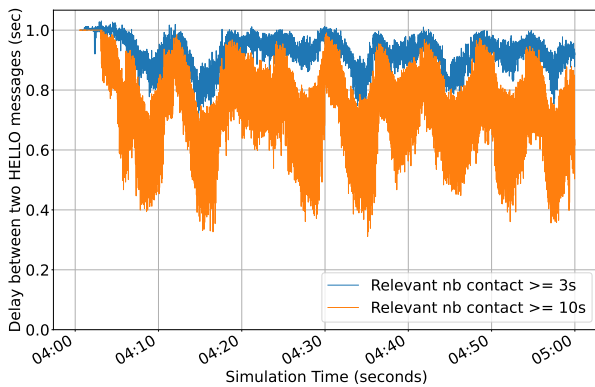


(c) Highway rush scenario.

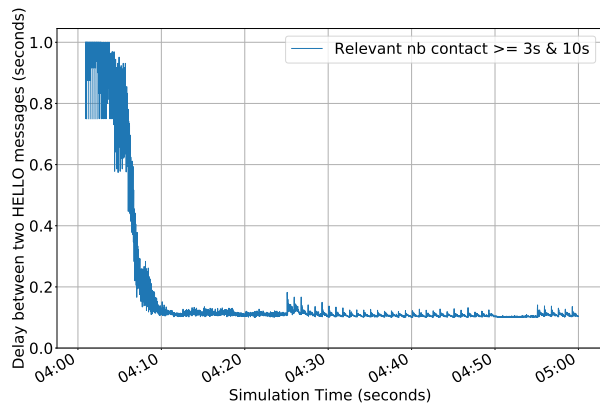


(d) Highway non-rush scenario.

Fig. 3: Proportion of relevant neighbours for each scenario.



(a) Evolution of d_{HELLO} in non-rush hour urban scenario.



(b) Evolution of d_{HELLO} in non-rush hour highway scenario.

Fig. 4: VANP behaviour

[3] F. Ingelrest, N. Mitton, and D. Simplot-Ryl, "A turnover based adaptive HELLO protocol for mobile ad hoc and sensor networks," in IEEE International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS), 2007.

[4] B. Kim, I. Kang, and H. Kim, "Resolving the unfairness of distributed rate control in the IEEE WAVE safety messaging," IEEE Transactions on Vehicular Technology, vol. 63, no. 5, pp. 2284–2297, 2014.

[5] N. Chaabouni, A. Hafid, and P. K. Sahu, "A collision-based beacon rate adaptation scheme (CBA) for VANETs," in IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS), 2013.

[6] G. Bansal, J. B. Kenney, and C. E. Rohrs, "Limeric: A linear adaptive message rate algorithm for dsrc congestion control," IEEE Transactions on Vehicular Technology, vol. 62, no. 9, pp. 4182–4197, 2013.

[7] G. Abbas, S. Ullah, M. Waqas, Z. H. Abbas, and M. Bilal,

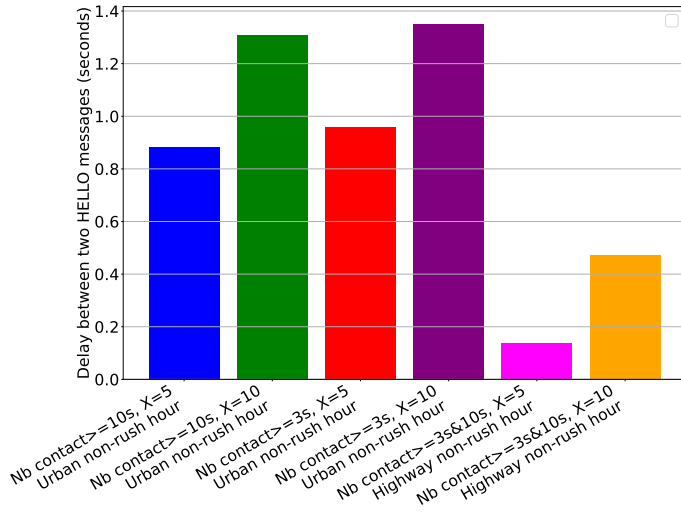
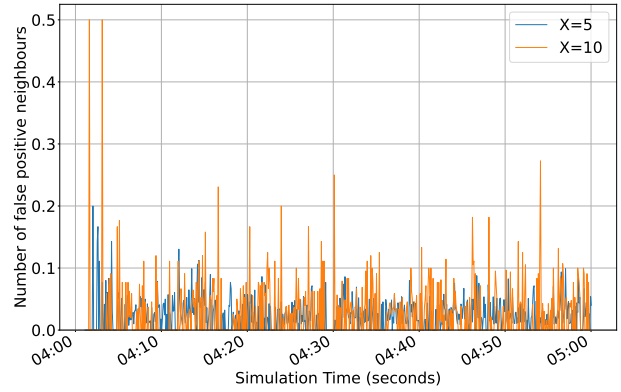


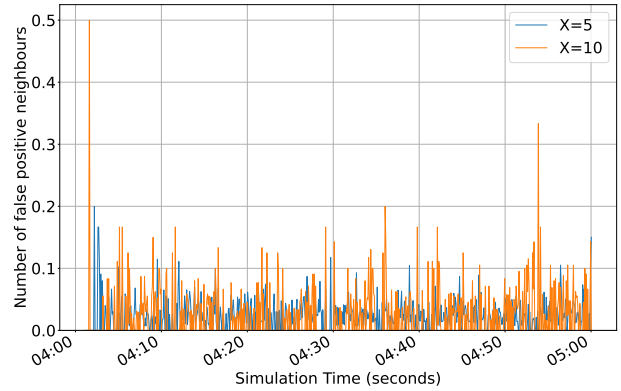
Fig. 5: Average d_{HELLO} in non-rush hour scenarios

“A position-based reliable emergency message routing scheme for road safety in VANETs,” *Computer Networks*, vol. 213, p. 109097, 2022.

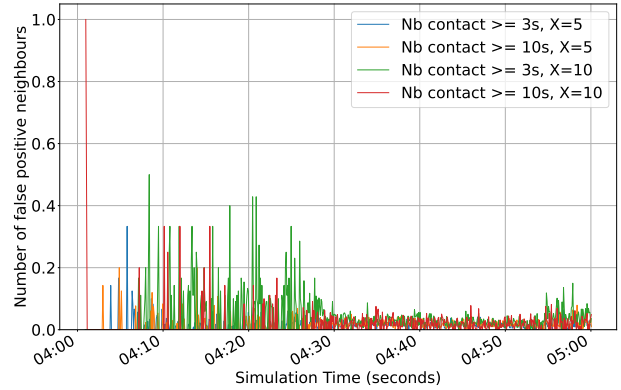
- [8] J. Li, “Cooperative beacon rate adaptation for vehicular ad-hoc network,” Ph.D. dissertation, KTH Royal Institute of Technology, 2021.
- [9] M. Wang, T. Chen, F. Du, J. Wang, G. Yin, and Y. Zhang, “Research on adaptive beacon message transmission power in vanets,” *Journal of Ambient Intelligence and Humanized Computing*, pp. 1–13, 2020.
- [10] F. Lyu, H. Zhu, N. Cheng, Y. Zhu, H. Zhou, W. Xu, G. Xue, and M. Li, “Abc: Adaptive beacon control for rear-end collision avoidance in vanets,” in 15th IEEE International Conference on Sensing, Communication, and Networking (SECON), 2018.
- [11] F. Li and C. Huang, “A mobility prediction based beacon rate adaptation scheme in VANETs,” in IEEE Symposium on Computers and Communications (ISCC), 2018.
- [12] H.-H. Nguyen and H.-Y. Jeong, “Mobility-adaptive beacon broadcast for vehicular cooperative safety-critical applications,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 6, pp. 1996–2010, 2017.
- [13] D. Barbieri, I. Thibault, D. Lister, A. Bazzi, B. M. Masini, and O. Andrisano, “Adaptive beaconing for safety enhancement in vehicular networks,” in 15th International Conference on ITS Telecommunications (ITST), 2017.
- [14] F. Goudarzi and H. Asgari, “Non-cooperative beacon rate and awareness control for VANETs,” *IEEE Access*, vol. 5, pp. 16 858–16 870, 2017.
- [15] P. A. Lopez, M. Behrisch, L. Bieker-Walz, J. Erdmann, Y. P. Flotterod, R. Hilbrich, L. Lucken, J. Rummel, P. Wagner, and E. Wiebner, “Microscopic Traffic Simulation using SUMO,” in IEEE Conference on Intelligent Transportation Systems (ITSC), 2018.
- [16] A. Varga and R. Hornig, “An Overview of the OMNeT++ Simulation Environment,” in 1st International Conference on Simulation Tools and Techniques for Communications, Networks and Systems (SIMUTools), 2008.
- [17] C. Sommer, R. German, and F. Dressler, “Bidirectionally Coupled Network and Road Traffic Simulation for Improved IVC Analysis,” *IEEE Transactions on Mobile Computing*, vol. 10, no. 1, pp. 3–15, 2011.
- [18] M. Gueriau and I. Dusparic, “Quantifying the impact of connected and autonomous vehicles on traffic efficiency and safety in mixed traffic,” in IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC), 2020.



(a) Urban scenario - Neighbours with contact duration greater than 3s

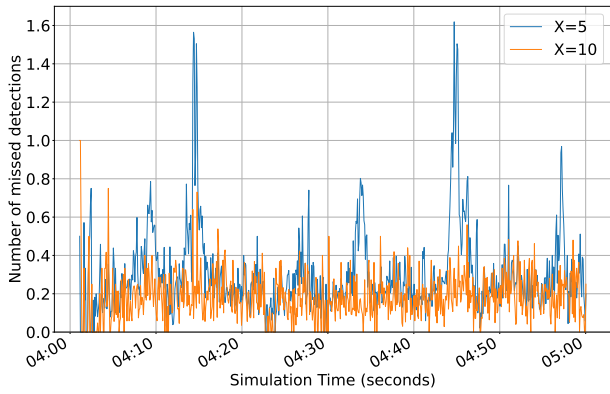


(b) Urban scenario - Neighbours with contact duration greater than 10s

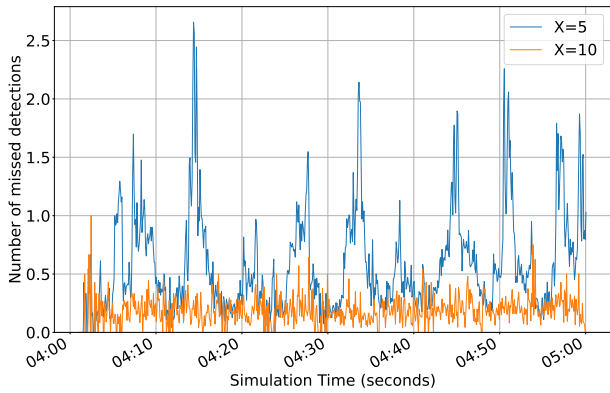


(c) Highway scenario

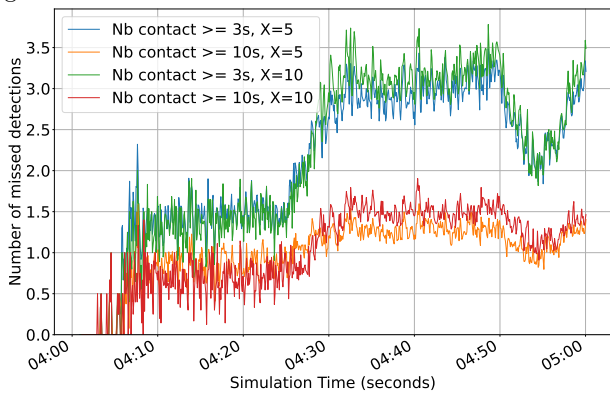
Fig. 6: Number of false positive neighbours in different non-rush hour scenarios



(a) Urban scenario - Neighbours with contact duration greater than 3s



(b) Urban scenario - Neighbours with contact duration greater than 10s



(c) Highway scenario

Fig. 7: Number of missed relevant neighbours in different non-rush hour scenarios