Modeling Efficient and Effective Communications in VANET through Population Protocols

Article

Accepted version

Bordonaro A., Concone F., De Paola A., Lo Re G., and Das S. K.

In Proceedings of the 2021 IEEE International Conference on Smart Computing (SMARTCOMP)

Publisher: IEEE

Modeling Efficient and Effective Communications in VANET through Population Protocols

Antonio Bordonaro[†], Federico Concone[†], Alessandra De Paola[†], Giuseppe Lo Re[†], and Sajal K. Das[≀]
[†]Department of Engineering, University of Palermo, Palermo, Italy
[≀]Department of Computer Science, Missouri University of Science and Technology, Rolla, USA
Emails: [†]{antonio.bordonaro,federico.concone,alessandra.depaola,giuseppe.lore}@unipa.it, [≀]sdas@mst.edu

***Abstract*—Vehicular Ad-hoc NETworks (VANETs) enable a countless set of next-generation applications thanks to the technological progress of the last decades. These applications rely on the assumption that a simple network of vehicles can be extended with more complex and powerful network infrastructure, in which several Road Side Units (RSUs) are employed to achieve application-specific goals. However, this assumption is not always satisfied as in many real-world scenarios it is unfeasible to have a conspicuous deployment of RSUs, due to both economic and environmental constraints. With the aim to overcome this limitation, in this paper we investigate how the only Vehicle-to-Vehicle (V2V) communications can be effectively exploited to share data among the vehicles about an event of interest, such as vehicular traffic. In this sense, we propose a novel communication schema based on the Population Protocol model that allows vehicles to be efficiently updated about a given event. Experimental analysis aims to evaluate the performance of the proposed schema, while also highlighting the benefits it might bring in VANETs applications.**

***Index Terms*—VANET, population protocols, Vehicle-to-Vehicle communications**

I. Introduction and State-of-the-Art

In recent years, Vehicular Ad-hoc NETworks (VANETs) [1] have captured the attention of the industrial and academic community as they provide solutions to address several real-life challenges [2], [3]. Communication between vehicles can occur through the *Road Side Unit (RSU) Communication Model*, the *Cluster Based Communication (CBC) Model* and the *Vehicle to Vehicle (V2V) Communication Model*.

In the *RSU Communication Model*, vehicles exchange information only through trusted entities called Road Side Units. Vehicles management and coordination is simplified due to the availability RSUs, but their pervasive deployment throughout the urban environment is a strong limitation in the adoption of this model, due to physical constraints and high deployment costs [4]. Furthermore, RSUs represent a single point of failure: if the RSU fails, the controlled vehicles can no longer communicate. In the *CBC Model*, vehicles are grouped into different *clusters*, each of which is identified by a special head vehicle. Communications between clusters occur only through the head vehicles which, subsequently, send information to its cluster vehicles. This architecture has several advantages, such as the possibility to adopt a communication schema independent from any infrastructure and a reduced network failure rate, but it introduces new limitations, e.g. the overhead required for the head election and the strong dependency of the intra-cluster communication on the connectivity with the cluster head.

For all these reasons, in recent years, great importance is given to the development of algorithms and protocols that use V2V communications that come up with several benefits compared with the other ones, e.g. the absence of an ad-hoc infrastructure [5]. Nevertheless, some limitations still need to be addressed, e.g. strong dependence on the adopted routing protocol and limited ability to adapt to sudden changes in network topology. Therefore, it is necessary the design of an efficient model that can, on the one hand, overcome the limitations of RSU- and CBC-based models and, on the other hand, fully exploit the potential of a Vehicle to Vehicle Communication Model.

In this paper, we propose a novel scheme for V2V communications based on the Population Protocol (PP), a theoretical model originally proposed to describe interactions in the wireless sensor networks. Here, it is assumed that a population is composed of vehicles that continuously interact in order to allow the population to converge on a common view of the monitored phenomenon and accomplish tasks typical of distributed systems, such as the counting problem [6], the majority problem [7], the reputation assignment problem [8], [9], or the leader election problem [10]. The correct definition of a given Population Protocol must exploit the features of interactions in the specific domain, which must be properly defined for the particular application field. For this reason, theoretical assumptions on which the population protocols are based, might not be fulfilled in a real scenario because of the model physical constraints, such as the speed of the vehicles [11] or transmission failures [12]. The deployment of a PP algorithm in a real context, therefore, requires an additional effort that ensures that basic assumptions of the model are fulfilled.

In essence, the contributions of this paper are as follows:

- We propose a communication scheme that enables the adoption of the Population Protocols model in a VANET scenario ensuring that the theoretical assumptions, required by the model to achieve convergence, are fulfilled.
- The proposed communication scheme ensures the vehicles to converge to the same information, and allows vehicles to assume different roles during the communication, thus enabling the implementation of asymmetric algorithms.

- The proposed scheme can support VANET-based applications in which vehicles have to be efficiently updated about specific events, e.g. vehicular traffic along the road.
- To the best of our knowledge, we are the first to apply the Population Protocol model to VANET applications. Our analysis suggests that our approach is suitable for real application scenarios, as it shows the best trade-off in terms of messages exchanged and convergence time.

This work is organized as follows. Section II introduces the general population protocol model as well as the requirements to adapt this model to VANETs. Then, the proposed approach is described in Section III, and it is followed by the experimental assessment in Section IV. Finally, Section V concludes this work and presents some future directions.

II. Preliminaries and Requirements

This section opens with an overview about Population Protocols, providing a solid starting point to fully comprehend the proposed approach. Then, the requirements to adapt this model to VANET applications are presented.

A. Background on population protocols

The Population Protocols (PPs) [13] model a distributed system as a population of interacting agents. Formally, a population is composed of N agents, with $N \geq 2$, each of which is defined as an automata with a finite number of states in a state space S. Each agent is initialized with an input value σ from an alphabet Σ that is used by an *input mapping function* $\lambda(\cdot)$ to set its initial state $s_i \in S$. Pairwise interactions update the states of both agents according to a *transition function* $\delta(\cdot)$ that takes both states as input and returns new states for both agents as output. Here, the agents' interactions are considered (i) unpredictable because there is no knowledge about the order in which the interactions occur, and (ii) asymmetric, i.e. one of the agents is the initiator of the interaction, and one the responder. Each agent is able to produce an output value that describes its own perception about the surrounding environment. This information is generated by using an appropriate *output mapping function* $\Omega(\cdot)$, which maps the current state $s_i \in S$ into a value $z \in Z$, where Z is the output alphabet. However, the node is unable to determine whether convergence has been achieved.

B. Requirements

The general formulation of the PP model does not cover constrains related to the specific application scenario. It is clear that the idea of agents that randomly interact fits the scenario addressed in this paper, i.e. a vehicular network populated by vehicles moving around an urban area. However, the model has to be extended to cover some aspects that may occur in a VANET-based application.

With the aim to better introduce these aspects, consider a scenario in which two vehicles, v_1 and v_2, with communication range r_1 and r_2 respectively, meet along their route. Assuming that $r_1 > r_2$, there is a possibility that, depending on the relative positions of the two nodes, only v_1 can send its state to v_2, but not vice versa: v_2 receives the state of v_1 and updates its state; while v_1 does not receive the information and, therefore, cannot update its state. This situation results in asymmetric communication, where only one node updates its state after an interaction. This is just one example of the numerous issues that can negatively affect communications between vehicles. In order to face these problems, the protocol we are proposing for VANETs should meet two requirements: *stable interaction* and *roles for asymmetric communications*.

The first requirements guarantees that interaction among vehicles occurs only when both interacting nodes have necessary information (i.e. the state of the other interacting node). This can be satisfied through the adoption of acknowledgments mechanism. The second requirement guarantees a mechanism to correctly manage asymmetric interactions among vehicles. In particular, the proposed protocol supports the presence of agents with different roles. Although PP models already include this feature, it is necessary to modify the underlying logic to prevent interacting agents from playing the same role.

III. Population Protocol for Vehicular Networks

In this section, we describe how the population protocols can be employed to fit the purpose of the referenced scenario. Then, we introduce the proposed communication protocol, from the finite-state machine to the messages used in the protocol.

A. The reference smart scenario

In this work, we are considering a scenario where N vehicles $\{v_1, \ldots, v_N\}$ are able to collect and share data about their surroundings in order to detect an excessive traffic condition by means of the measured speed thought On-Board-Units (OBUs), i.e., ad-hoc sensors embedded in vehicles or sensors contained in passengers' smart devices, such as GPS [14]. In general, a situation in which vehicles are traveling with low speed is very informative of vehicular traffic in a specific area.

The considered task can be fully accomplished by a population protocol algorithm that aims to know whether vehicles within a circumscribed geographic area are moving with high or low speeds. A possible approach consists in modeling it as a counting problem (refer to Section IV), which aims to calculate the difference between vehicles belonging to two different classes, i.e. vehicles with high speed and low speed. It is important to note that, in such a composite scenario, the over-traffic information must meet both geographic and time validity, because only vehicles belonging to a restricted geographic area can benefit from knowing about the event of interest [15]. A straightforward solution to address both issues could be, also here, to leverage on two Population Protocol models running in parallel to the proposed one. In particular, to address the first issue, it is possible to cluster the population of vehicles. In this case, a Population Protocol model requires the knowledge of information related to the geographic position of the vehicle, which can be retrieved by means of a GPS sensor embedded in the OBUs previously mentioned.

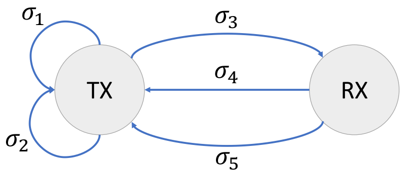


Fig. 1: The finite-state machine describing a vehicle.

Having this in mind, we can assume that the population is divided into clusters and that the algorithm disseminates the relevant information only within the cluster. This assumption allows the experimental evaluation to be focused in a scenario that consists of a single cluster. The extension to a multi-cluster scenario is trivial. It is sufficient that a vehicle is cluster-aware, so that it can ignore information from other clusters that would not be useful. This is easily achieved by extending the state structure to include cluster information.

Regarding the time validity of an event, we adopted the following solution, which allows to partially solve the well-known problem of vehicle synchronization. In particular, it is required that the Population Protocol algorithm be re-executed at regular intervals. Specifically, the operations performed by the vehicle v_i at each restart are as follows:

- stop of the Population Protocol algorithm;
- acquisition of information from the surrounding environment, through appropriate sensors. Specifically, speed is detected by performing the difference of two consecutive position measurements through GPS.
- use of logical predicates to establish, based on the information sensed from the environment, the initialization value σ_i and determine, through the input mapping function Ω, its initial state as $s_i = \Omega(\sigma_i)$. In our system we use the binary variable *traffic* that assumes $TRUE$ value if the vehicle speed is higher than a specific threshold, $FALSE$ otherwise;
- start of the Population Protocol algorithm and, therefore, the interactions with other vehicles.

This process requires only that vehicles have an awareness of the current time, with a level of approximation of even tens of seconds, a realistic situation in several real-world scenarios.

B. A finite-state machine for vehicles

We modeled each vehicle as a finite-state machine in which only two states are considered, as shown in Fig. 1.

Formally, these states represent the different roles a vehicle can assume: a vehicle is in the *transmitter state* (TX) when it starts the communication; a vehicle is in the *receiver state* (RX) when it first receives a message from another vehicle.

A vehicle always starts in the TX state by periodically *broadcasting a message* (σ_1) to other vehicles within its communication range. After an interaction occurs, it can update its state according to the role it assumes during the communication. The role decision mechanism is based on a a random identifier contained in the messages broadcasted by the two vehicles. In other words, the vehicle has *sent a message with a higher identifier than the one contained in the received message* (σ_2), then it remains in TX state; vice versa, if the vehicle has *sent a message with a lower identifier than the one contained in the received message* (σ_3), then it moves to the RX state. For the sake of clarity, we express this condition in the following way:

$$\delta(TX) = \begin{cases} RX & \text{if } \sigma_3 \text{ occurs} \\ TX & \text{otherwise.} \end{cases} \quad (1)$$

Finally, if during the communication a vehicle assumes the RX role, it has two possible options to return to TX. The first case, the simplest, occurs when a *timeout is reached during communication* (σ_4). This means that something gets wrong during the pairwise interaction between vehicles. Conversely, the second one occurs when *the communication is completed successfully* (σ_5). This means that regardless of the success or failure of the communication, the vehicle will always return to the TX state, ready to start a new communication.

TX Vehicle		*RX Vehicle*
$ID_{TX} \leftarrow getID()$		
$S_{TX} \leftarrow getState()$		
$ack \leftarrow 0$		
$M_1 = [ID_{TX}, S_{TX}, ack]$		
$M_1.send()$	$\longrightarrow$	$M_1 \leftarrow receive()$
		$ID_{RX} \leftarrow getID()$
		$S_{TX} \leftarrow M_1.get(S_{TX})$
		$S_{RX} \leftarrow getState()$
		$ack \leftarrow M_1.get(ID_{TX})$
		$M_2 = [ID_{RX}, S_{RX}, ack]$
$M_2.receive()$	$\longleftarrow$	$M_2.send()$
$ID_{TX} \leftarrow getID()$		
$S_{RX} \leftarrow M_2.getState()$		
$S_{TX} \leftarrow update(S_{TX}, S_{RX})$		
$ack \leftarrow M_2.get(ID_{RX})$		
$M_3 = [ID_{TX}, ack]$		
$M_3.send()$	$\longrightarrow$	$M_3 \leftarrow receive()$
		$S_{RX} \leftarrow update(S_{TX}, S_{RX})$

Fig. 2: Messages exchanged among two vehicles in the proposed three-way protocol.

C. The proposed three-way protocol

In a VANET-based scenario, for two vehicles to communicate properly, their roles prior to the communication must be established because it is impossible for them to assume the same role at the same time (refer to Section II). This situation occurs when, in the initial phase of the interaction, vehicles are simultaneously transmitting their message and receiving another message from other vehicles. This issue is faced by the finite-state machine summarized in Fig. 1, in which each vehicle simply checks for the random identifiers and applies Eq. 1. Having all this in mind, the resulting communication protocol proceeds as shown in Fig. 2.

During the broadcast phase, the TX vehicle has already sent the message M_1 containing a random identifier for the message ID_{TX}, a state S_{TX} representing the perception the vehicle have about the surrounding environment, and an acknowledgment ack initialized to zero. Here, the acknowledgment mechanism is used to be sure about the outcome

TABLE I: Parameters to model the counting problem [16].

Parameter	Value
Input Alphabet	$\Sigma = \{A, B\}$
Input mapping function	$\lambda(\sigma) = \begin{cases} M & \text{if } \sigma = A \\ -M & \text{if } \sigma = B \end{cases}$
Set of states	$S = \{-M, -M+1, ..., M-1, M\}$
Output mapping function	$\Omega(x) = \frac{nx}{m} + \frac{1}{2}$
Output Alphabet	$Z = \{-n, -n+1, ..., n-1, n\}$
Transition function	$f(a,b) = \begin{cases} (\frac{a+b}{2}, \frac{a+b}{2}) & \text{if } a+b \text{ is even} \\ (\frac{a+b-1}{2}, \frac{a+b+1}{2}) & \text{if } a+b \text{ is odd} \end{cases}$

TABLE II: Communication schemata used for the experiments.

Protocol	From → To	Messages
Naive	TX → TX	$M_1 : [state_{TX}]$
2-way	$TX \rightarrow RX$	$M_1 : [ID_{TX}, state_{TX}, ack = 0]$
	$RX \rightarrow TX$	$M_2 : [ID_{RX}, state_{RX}, ack = ID_{TX}]$
4-way	$TX \rightarrow RX$	$M_1 : [ID_{TX}, state_{TX}, ack = 0]$
	$RX \rightarrow TX$	$M_2 : [ID_{RX}, state_{RX}, ack = ID_{TX}]$
	$TX \rightarrow RX$	$M_3 : [ack = ID_{RX}]$
	$RX \rightarrow TX$	$M_4 : [ack = ID_{TX} + 1]$
3-way(Our)	$TX \rightarrow RX$	$M_1 : [ID_{TX}, state_{TX}, ack = 0]$
	$RX \rightarrow TX$	$M_2 : [ID_{RX}, state_{RX}, ack = ID_{TX}]$
	$TX \rightarrow RX$	$M_3 : [ack = ID_{RX}]$

of the communication. The vehicle RX receives the message and extracts the S_{TX}, necessary to update its current state in the last phase of communication. Then, it prepares M_2 by including the ack initialized with ID_{TX}, and send it to the transmitter vehicle. After receiving M_2, the vehicle TX knows the state of RX and runs the update state function. As final step, the vehicle TX sends a further acknowledge message to the vehicle RX in order to communicate the correct reception of its state. The reception of M_2, triggers the performing of the update state function by the vehicle RX.

Discussion: Since vehicles exchange messages through an unreliable channel, it is possible that some of the sent messages do not reach its destination. If M_1 is not transmitted correctly, the protocol does not start and, therefore, no vehicle updates its state. In this case, the system remains in a consistent state. If M_2 is lost, the two vehicles will stop the protocol after a certain time interval. No vehicle updates its state and the system remains in a consistent state. On the other hand, if the M_3 message is lost, the vehicle TX updates its state, but RX does not perform its state update, thus driving the system in an inconsistent state. Although the protocol may not reach an agreement between two vehicles, it is well-known in the literature the impossibility to define a communication protocol capable of reaching an unquestionable agreement between entities communicating over an unreliable channel. However, performed experimental evaluation proved that the proposed three-way protocol represents the best trade-off between accuracy and communication complexity, thus not further acknowledge messages are convenient.

IV. Experimental Evaluation

The experimental evaluation was conducted considering a scenario in which vehicles within a specific geographic area aim to exchange information about over-traffic conditions. Specifically, the proposed approach extracts information regarding the vehicular traffic based on the number of vehicles that are traveling at a certain speed. We are assuming that each vehicle has a sensor (embedded in vehicle or passengers' smart devices) that returns a boolean value, indicating whether the vehicle speed is above or below a certain threshold.

The task of determining the presence of vehicular traffic can be modeled through the counting problem, which general parameters for a PP-based approach are formally discussed in [16], and summarized in Table I.

The underlying logic is to start from a population of N agents, and calculate the difference k between the number agents of a class A and the number of agents of a class B, i.e. $k = N_A - N_B$. In this way, after T iterations, all the agents converge to the same value of k, according which it is possible to know if the majority is composed of vehicles of class A (or equivalently vehicles belonging to class B).

In the case of the addressed scenario, we consider that vehicles having speed higher than a given threshold belong to the class A, while the remaining ones belong to class B. Then, after a certain number of iterations, vehicles are simply asked to check for the value and, based on the majority, determine the presence of traffic in the nearby area.

Having all this in mind, the experimental section aims to compare the performance of the proposed protocol with other schemata characterized by a different exchange of messages, as summarized in Table II.

In the *Naive protocol* no acknowledgment messages are provided, thus leading in the impossibility to assign a role for the interacting vehicles because each vehicle updates its state when receiving a message.

In the *2-way protocol*, the receiver vehicle (RX) updates its state when it receives the initial broadcast message; while the sender vehicle (TX) updates its state when it receives an acknowledge corresponding to the last message sent.

The *4-way protocol* is similar to the proposed one, but it differs in the additional acknowledgment message required by the RX vehicle.

The remainder of this section describes all the experimental setup as well as the evaluation metrics used to perform a comparative analysis. Finally, a discussion about the achieved results is conducted.

A. Experimental Setup

All experiments were conducted using *VEINS*, a popular framework that simulates the road traffic with *SUMO* [17] and the network by employing *OMNET++*. Each simulation was performed using the following parameters.

Map: It consists of a grid of 21x21 two-lane roads, intersecting each other at a distance of 50 meters. Each side of the map thus has a size of $1km$, covering a territory of $1km^2$.

Beacon Interval [s]: It specifies the time interval between two consecutive messages during the broadcast phase (Message M_1). We set this value to 1s, which is the default value used for VANET simulations.

Number of Vehicles: During the experiments, this value was kept constant throughout the duration of the simulation. We set this value to 300.

Vehicles' communication range [m]: In our experiments, this value is 70m.

Speed Threshold [m/s]: It refers to the speed threshold that discriminates whether, according with the characteristics of the map considered, there is a congestion. In our experiments, it was set at a speed of 4 m/s.

Algorithm Reset Interval [s]: It represents the time between two consecutive resets of the algorithm, discussed in section III-A. It was set to 300s.

B. Metrics

In order to evaluate the performance of the adopted communication protocols, we used the metrics listed below.

Mean Square Error (MSE): The average squared difference between the estimated values and the actual value,

$$MSE = \frac{\sum_{i=0}^{N} (x - x_i)^2}{N} \tag{2}$$

Root Mean Square Error ($RMSE$): It represents the square root of MSE.

Number of Messages: It represents a measure of the protocol efficiency as it measures the total number of exchanged messages between the agents.

Mean Absolute Percentage Error ($MAPE$): It represents a *percentage error* calculated as a normalization of the output values error of the output values produced by the agents. Specifically, if k is the convergence value that should be produced by the algorithm, y_i is the output value of the i agent and N is the number of agents in the population,

$$MAPE = \frac{\sum_{i=0}^{N} |k - y_i|}{k \cdot N} \tag{3}$$

Error Function $\xi(t)$: This is an algorithm-dependent metric because it relates to an invariant property of the adopted Population Protocol algorithm. Considering the transition function shown in Table I, it is easy to prove that, in a theoretical scenario, the sum of the vehicles' states remains constant during the whole protocol execution. Therefore, for any time *t*, the following equation holds:

$$\sum_{i=1}^{N} vehicle_i.state(0) = \sum_{i=1}^{N} vehicle_i.state(t) \tag{4}$$

If some inconsistent state updates occur, the previous equation will not be satisfied. Then, we have defined the following error function,

$$\xi(t) = \left| \sum_{i=1}^{N} vehicle_i.state(0) - \sum_{i=1}^{N} vehicle_i.state(t) \right| \tag{5}$$

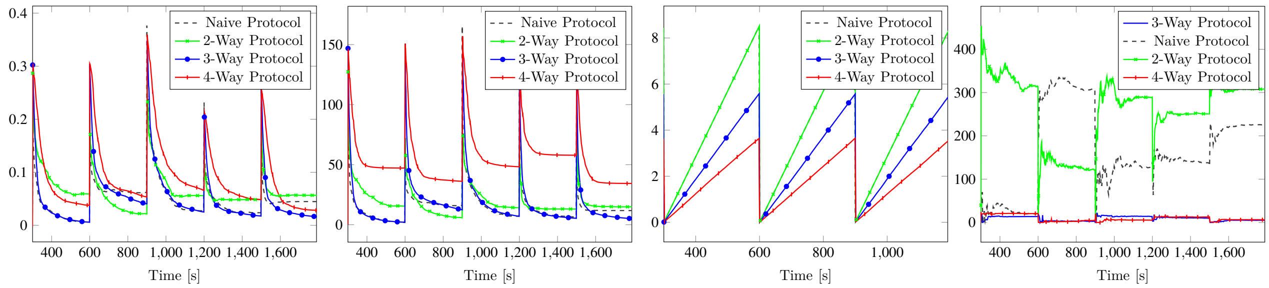
C. Experimental Results

The experimental evaluation of the proposed system provides analysis from both efficiency and accuracy perspectives. Performances related to the efficiency of the protocol were analyzed considering the amount of information exchanged between vehicles. It would be suitable that the protocol requires the exchange of a reduced number of messages to achieve convergence. In this regard, as can be seen from the Fig. 3c, the *4-way* schema achieves better results than other protocols. This is due to the fact that, in both the *3-way* and *4-way* protocols, once the vehicles start communicating (exchanging messages), the vehicles stop the broadcast process, thus reducing the number of sent messages. Since the *4-way* version involves the exchange of 4 messages (M_1, M_2, M_3, and M_4), the total time of interaction between vehicles will be longer and vehicles will spend more time without broadcasting messages. In fact, the *Naive* and *2-way* versions, which do not include any of these mechanisms, exchange even more messages, since the vehicles will always be in the broadcast phase. However, this has a significant impact on the convergence time.

In this regard, Fig. 3a, which allows to compare the Mean Absolute Percentage Error ($MAPE$) of the considered protocols, shows that the *3-way* protocol, performing interactions in a faster way, converges to a lower error value faster than the *4-way* version. Instead, as expected, the *Naive* and *2-way* versions, not implementing any type of control on the correct execution of the interaction, cause a greater number of incorrect transitions, resulting in the inconsistency of the system and producing incorrect output values. This is an expected result since the *Naive* and *2-way* versions do not implement mechanisms to ensure that the transition is performed correctly, as is done by the *3-way* protocol.

As for the specific error function ξ, in a theoretical case, it should always remain equal to 0. However, as can be seen from Fig. 3d, the ξ function has values differing from 0. Specifically, in the $Naive$ and *2-way* versions, which do not implement mechanisms to ensure that the interactions are successfully completed, this error has very high values. Instead, the *3-way* and *4-way* versions, which implement different control mechanisms, have significantly lower values, close to 0.

The *Naive* and *2-way* protocols achieve the lowest performance in both efficiency and correctness. Instead, the *3-way* and *4-way* schemata obtain comparable performances: both achieve similar results relative to the algorithm-dependent error function ξ however, while *4-way* obtains better results relative to the number of exchanged messages, *3-way* presents better performances relative to the convergence time. Due to the nature of the scenario and the application requirements, the reduction in the number of messages exchanged does



(a) MAPE (b) RMSE (c) Total number of Messages (d) $\xi(t)$ Error Function

Fig. 3: Performance of the considered communication protocols, specifically (a) shows the $MAPE$ values, (b) plots the $RMSE$ values, (c) shows the number of messages exchanged by the vehicles and (d) instead reports the values of the ξ Error Function.

not justify the decrease in convergence time performance. In fact, it is preferable to adopt a protocol that requires a larger amount of information exchanged between vehicles but produces correct results faster. For these reasons, the *3-way* protocol represents the best trade-off between the compared alternatives

V. Conclusion

In this paper, we discussed the possibility of adopting the Population Protocols model in a real-world scenario, where the theoretical assumptions required for convergence are not guaranteed. Specifically, we considered the VANET scenario and proposed an approach based on the Population Protocol model for propagating relevant events among the VANET vehicles. Although the properties of the Population Protocols model perfectly fit the characteristics of VANETs, we have identified two main problems that need to be solved to achieve our goals. The first relates to the consistent updating of states, while the second relates to the definition of different roles during the interaction between vehicles. We have addressed both of these issues by defining an appropriate *3-way* communication scheme. Experimental evaluation shows that the *3-way* protocol represents the best trade-off between computational complexity and accuracy. As future work, we want to study the management of agent synchronization. It would be desirable to achieve a fully asynchronous protocol capable of supporting dynamic information. Moreover, although there are population protocols capable of creating clusters of agents with similar characteristics, these have never been applied to the contexts of VANETs. Hence, we want to better investigate this aspect in order to make the proposed protocol as complete as possible. Finally, we plan to investigate the use of incentive mechanisms [18] to increase users' willingness to share information.

References

[1] T. K, S. T. M, B. M, and A. K. H, "Article: A survey on vanet technologies," *International Journal of Computer Applications*, vol. 121, no. 18, pp. 1–9, July 2015.

[2] H. Vasudev, V. Deshpande, D. Das, and S. K. Das, "A lightweight mutual authentication protocol for v2v communication in internet of vehicles," *IEEE Transactions on Vehicular Technology*, vol. 69, 2020.

[3] V. Agate, F. Concone, and P. Ferraro, "Wip: Smart services for an augmented campus," in *2018 IEEE International Conference on Smart Computing (SMARTCOMP)*, 2018, pp. 276–278.

[4] M. Al Shareeda, A. Khalil, and W. Fahs, "Towards the optimization of road side unit placement using genetic algorithm," in *2018 International Arab Conference on Information Technology (ACIT)*. IEEE, 2018.

[5] S. Yousefi, M. S. Mousavi, and M. Fathy, "Vehicular ad hoc networks (vanets): Challenges and perspectives," in *2006 6th International Conference on ITS Telecommunications*, 2006, pp. 761–766.

[6] P. Berenbrink, D. Kaaser, and T. Radzik, "On counting the population size," in *Proceedings of the 2019 ACM Symposium on Principles of Distributed Computing*, ser. PODC ´19. New York, NY, USA: Association for Computing Machinery, 2019, p. 43–52.

[7] P. Berenbrink, R. Elsässer, T. Friedetzky, D. Kaaser, P. Kling, and T. Radzik, "A population protocol for exact majority with $o(\log^{5/3} n)$ stabilization time and asymptotically optimal number of states," 2018.

[8] V. Agate, A. De Paola, G. Lo Re, and M. Morana, "Vulnerability evaluation of distributed reputation management systems," ser. VALUETOOLS'16. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2017, p. 235–242.

[9] V. Agate, A. De Paola, S. Gaglio, G. Lo Re, and M. Morana, "A framework for parallel assessment of reputation management systems," ser. CompSysTech '16. New York, NY, USA: Association for Computing Machinery, 2016, p. 121–128.

[10] D. Doty and D. Soloveichik, "Stable leader election in population protocols requires linear time," *Distributed Computing*, vol. 31, 2018.

[11] R. Sadano, Y. Sudo, H. Kakugawa, and T. Masuzawa, "A population protocol model with interaction probability considering speeds of agents," in *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 2019, pp. 2113–2122.

[12] G. A. Di Luna, P. Flocchini, T. Izumi, T. Izumi, N. Santoro, and G. Viglietta, "Population protocols with faulty interactions: the impact of a leader," *Theoretical Computer Science*, vol. 754, pp. 35–49, 2019.

[13] D. Angluin, J. Aspnes, Z. Diamadi, M. J. Fischer, and R. Peralta, "Computation in networks of passively mobile finite-state sensors," *Distributed computing*, vol. 18, no. 4, pp. 235–253, 2006.

[14] F. Concone, P. Ferraro, and G. Lo Re, "Towards a smart campus through participatory sensing," in *2018 IEEE International Conference on Smart Computing (SMARTCOMP)*, 2018, pp. 393–398.

[15] A. De Paola, P. Ferraro, S. Gaglio, and G. Lo Re, "Context-awareness for multi-sensor data fusion in smart environments," in *AI*IA 2016 Advances in Artificial Intelligence*. Cham: Springer International Publishing, 2016, pp. 377–391.

[16] Y. Mocquard, E. Anceaume, J. Aspnes, Y. Busnel, and B. Sericola, "Counting with population protocols," in *2015 IEEE 14th International Symposium on Network Computing and Applications*. IEEE, 2015.

[17] P. A. Lopez, M. Behrisch, L. Bieker-Walz, J. Erdmann, Y.-P. Flötteröd, R. Hilbrich, L. Lücken, J. Rummel, P. Wagner, and E. Wießner, "Microscopic traffic simulation using sumo," in *The 21st IEEE International Conference on Intelligent Transportation Systems*. IEEE, 2018.

[18] F. Restuccia, P. Ferraro, S. Silvestri, S. K. Das, and G. L. Re, "Incentme: Effective mechanism design to stimulate crowdsensing participants with uncertain mobility," *IEEE Transactions on Mobile Computing*, vol. 18, no. 7, pp. 1571–1584, 2019.