2021 20th International Conference on Ubiquitous Computing and Communications (IUCC/CIT/DSCI/SmartCNS)

VPP: A *Communication Schema* for Population Protocols in VANET

Antonio Bordonaro, Alessandra De Paola, Giuseppe Lo Re
Department of Engineering, University of Palermo, Palermo, Italy
Emails: {antonio.bordonaro, alessandra.depaola, giuseppe.lore}@unipa.it

***Abstract*—Population protocols are a new paradigm that aims to model distributed systems composed of randomly interacting mobile agents. The main advantage of this model is that, when certain theoretical assumptions hold, it is possible to formally demonstrate their convergence. We consider the problem of adopting Population Protocols in a real-world scenario that does not guarantee these assumptions. In particular, we consider the application domain of Vehicular Ad-Hoc Networks (VANETs) which are characterized by dynamic network topologies, and where wireless communications can be affected by interferences and errors. In this work, we analyze the main features of VANET agents and the communications between them, and how these features can affect the performance of a Population Protocol. Based on the obtained results, we propose a communication schema that allows to preserve the formal properties of Population Protocols. Experimental results prove the suitability of our approach, regardless of the specific scenario considered.**

I. INTRODUCTION

In recent years, Vehicular Ad-Hoc Networks (VANETs) [1], have attracted the attention of academic and industry researchers, as they provide a valid and unconventional way to address many issues in this research area. Communication between vehicles can take place through a vehicle-to-vehicle (V2V) or vehicle-to-infrastructure (V2I) architecture. The latter model relies on a network of reliable RSUs (Road Side Units) and allows for more complex services to be provided, since cloud-connected RSUs can leverage online computational resources. However, it should be considered that in most cases it may not be realistic to deploy RSUs in the entire area of interest. For these reasons, great importance is given to the development of algorithms and protocols that use V2V communications and exclusively exploit information and data received from other vehicles [2].

To fully exploit the potential of V2V, its many challenges need to be adequately addressed, including high node mobility and rapidly changing network topology. For these reasons, the main research direction concerns the implementation of a communication model that allows obtaining high performances in terms of Packet Delay, Packet Drop Rate, and Processing Time Performance [3].

To design an efficient communication schema in VANETs that takes advantage of a V2V architecture, we propose to adopt Population Protocols (PP), which were originally proposed to model wireless sensor networks, and which can be successfully used to model other distributed systems. The PP model states that a population of agents can converge towards a common value, by carrying out continuous interactions between pairs of agents selected at random, if each interaction causes the internal state of the interacting agents to be updated.

Nevertheless, many theoretical assumptions on which the PPs are based may not be satisfied in a real application scenario. Therefore, the design of a PP algorithm in a real context requires additional effort to ensure that the basic assumptions of the model are satisfied.

- Allows the execution of algorithms based on population protocols, ensuring that the theoretical assumptions required by the population protocol model are satisfied even in a real scenario such as VANET;
- Defines two roles, thus supporting the execution of asymmetric Population Protocols algorithms;
- Allows nodes to exchange information efficiently by leveraging V2V communications only.

The remainder of this paper is organized as follows. Section II describes the main applications of VANETs and PPs. Section III provides a brief description of the PP model. Section IV presents our communication schema by introducing the underlying constraints and the proposed solutions. Section V presents the experimental evaluation and finally, section VI contains conclusions and suggestions for future extensions.

II. RELATED WORK

VANETs [1] represent a special case of Mobile Ad-hoc Networks (MANETs) and play a key role in Intelligent Transportation Systems (ITSs) [4]. Due to the nature of the vehicular network, one of the main challenges still open concerns the efficient communication between heterogeneous and mobile vehicles [5].

The V2V communication model is the most promising for the development of VANET applications; however, it has some limitations that weaken its application. One of these is the strong dependence on the performance of the adopted routing protocol and a limited ability to adapt to sudden changes in the network topology [6].

Therefore, an efficient solution is needed to overcome the limitations of RSU-based models and fully exploit the potential of the V2V communication model. Here, we propose a communication schema that allows the design of VANET applications using the distributed model of PPs [7].

The PP model provides an effective formalism to address typical problems of distributed systems. For example, the authors of [8] addressed the counting problem, while the authors of [9] proposed a solution to the majority problem, while the

978-1-6654-6667-7/21/$31.00 ©2021 IEEE
DOI 10.1109/IUCC-CIT-DSCI-SmartCNS55181.2021.00017

authors of [10] addressed the leader election problem. The original PP model suffers from some expressive limitations, solved by further extensions proposed in subsequent works, such as Communication Protocols [11], which extend PPs by introducing identifiers, or Mediated Protocols [12], which introduce states associated with the network edge.

The PP model requires theoretical assumptions that in many real-world scenarios may not be satisfied. In this regard, some works extend the model by taking into account constraints related to physical world restrictions. For example, the authors of [13] introduced the possibility to modify the speed of the nodes to evaluate the effect on the probability of interaction and on the convergence time, while the authors of [14] analyzed the effect of transmission faults.

III. POPULATION PROTOCOL

Population Protocols [7] model a distributed system as a population of interacting agents. Each agent is initialized with an input value which determines its initial state, according to an input mapping function. The interactions occur on a pairwise basis and each interaction causes the updating of the state of the interacting nodes according to a transition function that takes their current states as input. When the PP halts, each agent produces an output value based on an output mapping function, which accepts the node state as input.

Typically, algorithms based on the Population Protocol model are characterized by the following features [15]:

- **Agent finite-state machine**: each agent can store a finite number of bits, which does not depend on population size.
- **Uniformity**: the algorithm does not depend on the population size.
- **Computation based on interactions**: the only way to update agent states is through interactions. The concept of interaction must be defined according to the specific application field.
- **Unpredictability of interactions**: The order in which agents interact is random. However, for the convergence of the algorithm, the interactions must respect the *equity* constraint.
- **Distributed Input and Output**: The input and output values are provided and produced in a distributed way.
- **Convergence rather than termination**: agents are generally unable to determine when the algorithm reaches the convergence.

Formally, a population protocol-based algorithm is defined by:

- Σ: a finite sequence of symbols that can be provided as input to the agents during the initialization phase of the algorithm;
- S: a finite set of agent states;
- $\lambda(\sigma)$: input mapping function that maps each $\sigma \in \Sigma$ element into an $s \in S$ element. It determines the initial state of agents;
- Z: a finite sequence of symbols that an agent can produce as an output value;
- $\Omega(s)$: an output mapping function that maps each element $s \in S$ into an element $z \in Z$. It allows each agent to determine its output value z, based on its current state s;
- $\delta(s_1, s_2) \subseteq S^4$: a transition function that accepts as input the states of two interacting agents and returns the new pair of states.

The algorithm is initialized by providing each node with an input value σ and, using the $\lambda(\sigma)$ function, the agents determine the initial state. The nodes can then start interacting with each other, changing states according to the $\delta(s_1, s_2)$ transition function. Each node will produce an output value through the output mapping function $\Omega(s)$.

IV. VPP: A POPULATION PROTOCOL FOR VANET

The specific scenario we considered, i.e. a vehicular network populated by agents representing moving vehicles, has some differences compared to the abstract model of Population Protocols, thus it is necessary to introduce the following two considerations:

- **Consistent update of states**: in the theoretical model of population protocols, nodes are interconnected by defining an interactions graph, which is assumed to be complete (all nodes are interconnected with each other), and the interactions occur randomly. Also, any interaction between nodes is assumed to be successful (both nodes update their states consistently). Unlike the theoretic model, in a real VANET, this assumption cannot be guaranteed. For instance, interferences, communication errors, physical obstacles, different communication ranges can produce two different results in interacting agents. Without the guarantee that the status update is consistent, the algorithm may not behave correctly and produce the correct output values.
- **Different roles of agents involved in the interaction**: In the theoretic model, the communication protocol allows nodes to assume different roles. Specifically, if there are two different roles, i.e., *role A* and *role B*, the transition function can be expressed as follows:

  $$(a', b') = f(a, b) \rightarrow \begin{cases} a' = f_A(a, b) & \text{if } \textit{role A} \\ b' = f_B(a, b) & \text{if } \textit{role B} \end{cases} \tag{1}$$

  Successful implementation of this transition rule requires nodes to be able to determine their role in the communication, to take appropriate action. That is, if both nodes assumed the same role, their state would not be updated correctly.

The first point states that an interaction between two nodes should only occur when they both have all the necessary information, i.e., the state of the other node. This may not happen if a transmission error occurs during the state exchange or if nodes have different communication ranges. If only a node updates its state, the interaction cannot be considered correct and the algorithm will produce incorrect output values. The PP for VANET (VPP) proposed here addresses this problem through the adoption of acknowledgment messages.

Regarding the second point, VPP must support different agent roles, ensuring that two interacting nodes never play the same role. Thus, our solution supports model asymmetric communications, distinguishing between the vehicle that acts as the transmitter node (TX) and the other that acts as the receiver (RX).

The resulting communication protocol proceeds as represented by the finite state machine shown in Fig. 1. Each node periodically broadcasts a message (state TX in Fig. 1) containing its state value. During this broadcast phase, a node can receive state messages from other nodes within its communication range. To introduce the communication asymmetry necessary to perform the state update described above, each node selects a random identifier for each first message, and it plays the RX role only if it receives a state message from a node with a higher identifier (see the transition from state TX to state RX in Fig. 1). In this case, the RX node replies with an acknowledge message to the TX node. A node plays the TX role when it receives the acknowledge message from another node. At this point, the TX role knows the other node's state and can perform the update state function corresponding to its role. As a final step, the TX node sends a further acknowledge message to the RX node to communicate the correct reception of its state. The receipt of this message triggers the execution of the state update function by the RX node. The resulting sequence of messages is summarized in Fig. 2.

Since nodes exchange messages through an unreliable channel, it is possible that some of the sent messages do not reach their destination. If the M1 message is not transmitted correctly, the protocol does not start and, therefore, no node updates its state. In this case, the system remains in a consistent state. If the M2 message is lost, the two nodes will stop the protocol after a certain time. No nodes update their state and the system remains in a consistent state. If, on the other hand, the M3 message is lost, the TX node updates its state, but the RX node does not perform its state update, thus bringing the system to an inconsistent state.

It is worth noticing the well-known impossibility to define a communication protocol capable of reaching an indisputable agreement between two nodes that communicate over an unreliable channel. The experimental evaluation carried out in this work showed that the proposed three-way protocol represents the best trade-off between accuracy and communication complexity, thus no further acknowledge message is convenient. Finding the right trade-off is also relevant considering that the devices installed in vehicles can have limited resources [16], [17].

A. VPP Message Structure and Pseudocode

VPP defines the structure of a unique message, used to implement all the messages of the three-way protocol. The message contains the following fields:

- **ID**: random message identifier;
- **ACK**: acknowledgment value equal to the identifier of the received message, or equal to 0 for the first message;
- **Payload Length**: an integer which represents the number of bytes of the payload field;
- **Payload**: Variable-length field that contains information required for the specific population protocol. Typically, it contains the state of the sender node.

The behavior of the VPP agent can be modeled through pseudocode 1.

1 - VPP Pseudocode

```
Broadcast
  this.isWaitingAck = False;
  while $True$ do
      /* TX-1 */
      M1=createM1(randomID, node.state, ack=0)
      send(M1)
  end while
On event: rcvPkt=recv()
  if ($rcvPkt.ack == 0 \wedge rcvPkt.id > M1.id \wedge \neg this.isWaitingAck$) then
      /* RX-1 */
      stopBroadcast();
      recvM1 = rcvPkt;
      M2 = createM2(randomID, node.state, ack=recvM1.id);
      send(M2);
      this.isWaitingAck = True;
      startTimer();
  else if ($rcvPkt.ack == M1.id$) then
      /* TX-2 */
      rcvM2 = rcvPkt;
      M3 = createM3(randomID, ack=rcvM2.id);
      send(M3);
      updateStateTX(node.state, rcvM2.state);
      startBroadcast();
  else if ($rcvPkt.ack == M2.id \wedge this.isWaitingAck$) then
      /* RX-2 */
      StopTimer();
      updateStateRX(node.state, rcvM1.state);
      startBroadcast();
      this.isWaitingAck = False;
  end if
On event: Timeout()
  this.isWaitingAck = False;
  stopTimer();
  startBroadcast();
```

B. Impact of Node Mobility

Due to node mobility, errors, and interferences in wireless communications, as previously noted, some inconsistent state transitions can occur. A state transition is defined as inconsistent if the state update function is performed only by a single node, rather than by a pair of nodes.

In wireless communications, the *Packet Error Rate* (PER) is defined as the percentage of packets that are corrupted during

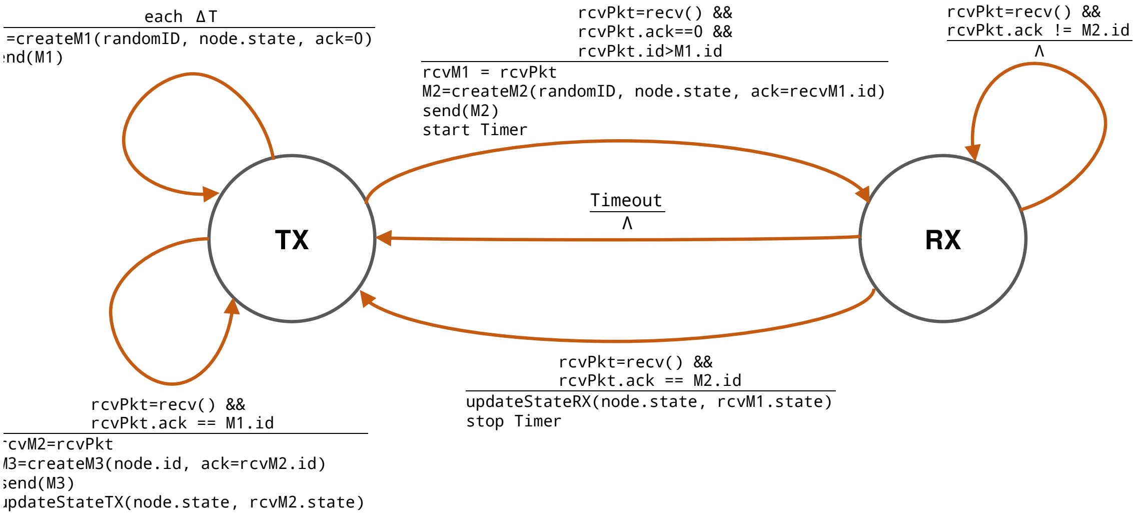


Fig. 1: VPP Finite state machine.

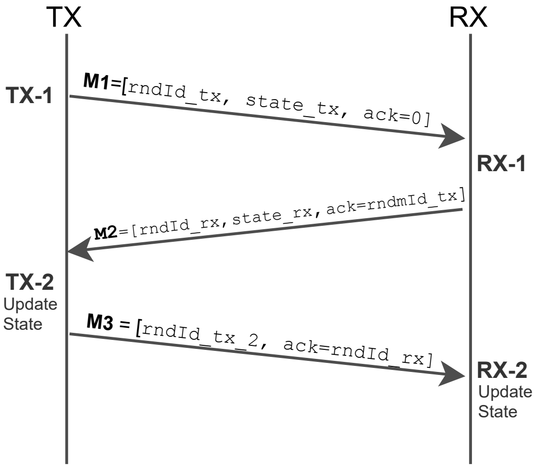


Fig. 2: VPP Sequence diagram.

a transmission [18]. The expected value of PER represents the *Packet Error Probability* for a packet M and is defined as:

$$p_p(M) = 1 - (1 - p_e)^{|M|} = 1 - e^{|M| log(1-p_e)}, \tag{2}$$

where $|M|$ is the number of bit of M, and p_e is the *bit error probability*. The value of p_e depends on the characteristics of the communication channel, such as noise or distortion.

Another problem influencing VPP is the physical arrangement of nodes. The distance d between two vehicles affects packet transmission due to signal attenuation. To provide a general model, we introduce the function $\gamma(d)$, which models the probability at least one bit of the message is corrupted, during a transmission occurring at a distance d. A suitable distance for transmission can range between 0 (ideally) and R (the communication range of the OBU). The distance varies within the following range:

$$\begin{gathered} \gamma(d) \in [\gamma_{min}; \gamma_{max}] \subseteq [0; 1], \\ \text{where } \gamma_{min} = \gamma(0) \text{ and } \gamma_{max} = \gamma(R). \end{gathered} \tag{3}$$

Thus, the probability of having a correct transmission has also to take into account the probability of not observing a message corruption due to the distance between the vehicles, i.e. $(1 - \gamma(d))$.

Finally, a third factor that could affect the transmission is the specific probability of faults due to hardware malfunctions. It is worth noticing that different vehicles can have different probabilities of hardware failure. If φ_x is the probability that a hardware fault will occur during the transmission of a message performed by the vehicle x, the probability of having a correct transmission has also to take into account the factor $(1 - \varphi_x)$.

By combining these factors, we can model the probability that a packet M, sent by the vehicle x, does not reach its destination correctly, as follows:

$$P_\epsilon(M, x) = 1 - (1 - p_p(M))(1 - \gamma(d))(1 - \varphi_x). \tag{4}$$

As described above, the only condition in which the system performs an inconsistent transition occurs when the vehicles A and B successfully exchange the messages $M1$ and $M2$, but an error occurs during the transmission of the message $M3$. The probability of such an event occurring can be expressed as follows:

$$P_{inc}(A, B) = \begin{cases} P_\epsilon(M_3, A) \text{ if } A.randId > B.randId \\ P_\epsilon(M_3, B) \text{ otherwhise} \end{cases} \tag{5}$$

where

$$\begin{cases} P_\epsilon(M_3, A) = P_\epsilon(\neg M_1, A) \cdot P_\epsilon(\neg M_2, B) \cdot P_\epsilon(M_3, A) \\ P_\epsilon(M_3, B) = P_\epsilon(\neg M_1, B) \cdot P_\epsilon(\neg M_2, A) \cdot P_\epsilon(M_3, B) \end{cases} \quad (6)$$

Some further considerations can be made about how the distance between a pair of vehicles varies. We can model their mutual movement, by assuming that for small time interval nodes move along their joining line. Let d_1, d_2 and d_3 be the values of the distance between the pair of nodes when M_1, M_2 and M_3 are transmitted. If vehicles move in the same direction, for a small time interval, we can assume that their distance is constant, i.e., $d_1 \approx d_2 \approx d_3$. If, on the other hand, the vehicles move in opposite directions, even at low speeds, their mutual distance varies during the exchange of messages, i.e. $d_1 \neq d_2 \neq d_3$. This difference can affect the probability of missing the last message, thus making an inconsistent transition.

Let d_0 be the distance between a pair of vehicles, A and B, at the time of first contact. Their position along the joining line depends on their speed, i.e., v_A and v_B. If we consider the position of A at the time of first contact as the origin of a relative reference system, A and B positions are expressed as follows:

$$\begin{cases} x_A(t) = \ \ v_A \cdot t \\ x_B(t) = -\, v_B \cdot t + d_0 \end{cases} \quad (7)$$

Consequently, their relative distance can be expressed as follows:

$$\begin{aligned} d(t) &= |x_A(t) - x_B(t)| = \\ &= |v_A \cdot t + v_B \cdot t - d_0| = \\ &= |(v_A + v_B) \cdot t + d_0| \end{aligned} \quad (8)$$

If we assume that the distance d_0 also represents the limit distance beyond which the two vehicles will no longer be able to communicate, the total contact time, i.e., the time interval during which the two vehicles can communicate, can be expressed as:

$$\Delta T_c = \frac{2d_0}{(v_A + v_B)}. \quad (9)$$

To achieve a successful state transition, the three-way message exchange must take less than ΔT_c time. Moreover, the higher the vehicle speeds, the more likely it is to observe inconsistent state transitions, as the time required to perform the three-way message exchange would be close to or greater than ΔT_c.

V. EXPERIMENTAL EVALUATION

To evaluate the suitability of VPP as a communication schema to be adopted to perform Population Protocols on VANETs, we compared it with other communication schemes characterized by a different number of messages. In particular, we considered, the following three protocols:

- **Naive Protocol**: no acknowledgment messages are provided; each node broadcasts its state and updates its state when it receives a message from another node; no different roles are defined for nodes involved in a communication.
- **2-way Protocol**: two roles are defined for nodes involved in a communication, i.e., TX and RX, as in VPP. The RX node updates its state when it receives the initial broadcast message from another node, while the TX node updates its state when it receives an acknowledge corresponding to the last message sent. The 2-way Protocol messages exchange is therefore defined as follows:

$$M1)\ Tx \rightarrow Rx : [rndId\_tx, state\_tx, ack = 0]$$

(RX updates its state)

$$M2)\ Rx \rightarrow Tx : [rndId\_rx, state\_rx, ack = rndId\_tx]$$

(TX updates its state)

- **4-way Protocol**: requires an additional acknowledgment message with respect to VPP, as follows:

$$M1)\ Tx \rightarrow Rx : [rndId\_tx, state\_tx, ack = 0]$$

$$M2)\ Rx \rightarrow Tx : [rndId\_rx, state\_rx, ack = rndId\_tx]$$

$$M3)\ Tx \rightarrow Rx : [ack = randId\_rx]$$

(RX updates its state)

$$M4)\ Rx \rightarrow Tx : [ack = randId\_tx + 1]$$

(TX updates its state)

The remainder of this section describes a specific algorithm adopted as a case study, the evaluation metrics used to perform our comparative analysis, specifies the experimental settings, and finally described the results obtained.

A. Case Study - the Counting Problem

The Counting Problem is a specific case of the network-size estimation problem and represents one of the most relevant problems in opportunistic networks.

In this paper, we consider the counting algorithm presented in [19] as a case study for evaluating the performance of the VPP communication schema. The authors of [19] model a counting algorithm through a Population Protocol and formally demonstrate its properties and convergence time.

According to this model, the population consists of N agents. Each agent is initialized with an input symbol $\sigma \in \{A, B\}$. The algorithm goal is to estimate the value $k = N_A - N_B$, where N_A is the number of nodes initialized with A, and N_B is the number of nodes initialized with B.

Node initialized with A set their initial state with a positive number M, while nodes initialized with A set their initial state with the negative number $-M$. This protocol updates the state of two interacting nodes with the average of their state values before the transition.

Formally, the algorithm is characterized by the parameters described in Table I.

B. Evaluation Metrics

In order to evaluate the performance of different communication protocols, we propose to adopt a set of metrics that are independent of the specific algorithm implemented and a set of metrics that rather depend on it. This latter set of metrics must be defined for each different algorithm considered.

We adopt the following algorithm-independent metrics:

- **Mean Absolute Percentage Error (MAPE)**: defined as the normalized average of the absolute error made by nodes. If x is the correct value that should be produced by the algorithm and x_i is the output value from the i node, the MAPE is defined as follows:

$$MAPE = \frac{\sum_{i=0}^{N} |x - x_i|}{x \cdot N} \tag{10}$$

- **Mean Square Error (MSE)**: defined as the average of the square error made by nodes:

$$MSE = \frac{\sum_{i=0}^{N} (x - x_i)^2}{N} \tag{11}$$

- **Root Mean Square Error (RMSE)**: defined as follows:

$$RMSE = \sqrt{MSE} \tag{12}$$

- **Number of Packets**: the total amount of packets transmitted during an experimental run.

The algorithmic-dependent metric we propose to adopt concerns an invariant property of the Population Protocol considered as a case study. According to the transition function shown in Tab. I, it is possible to prove that the sum of node states remains constant during the entire protocol run. That is, for any time t, the following equation holds:

$$\sum_{i=1}^{N} node_i.state(0) = \sum_{i=1}^{N} node_i.state(t) \tag{13}$$

If some inconsistent state updates are performed (i.e., performed by a single node of an interacting node pair), the eq. 13 will not be satisfied. We, therefore, adopt the following error function:

$$\xi(t) = \left| \sum_{i=1}^{N} node_i.state(0) - \sum_{i=1}^{N} node_i.state(t) \right| \tag{14}$$

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 I: Paremeters of the Population Protocol which models the counting algorithm.

C. Experimental Settings

The performance evaluation was conducted through simulation, by adopting some well-known simulation tools, such as the SUMO/VEINS simulator and the OMNET++ libraries.

Each simulation is specified by the following parameters:

- **Map**: a synthetic map composed of a network of streets. We consider two different sizes for maps:
  - **Big Map**: a grid of 16x16 streets, which intersect each other, with a distance of 100 meters between two consecutive parallel streets. The map covers an area of 2.25 km^2, with 25.6 km of roads.
  - **Small Map**: a grid of 8x8 streets, which intersect each other, with a distance of 100 meters between two consecutive parallel streets. The map covers an area of 0.49 km^2, with 6.4 km of roads.
- **Vehicle density**: during simulation, we considered two different density levels:
  - **High Density**: 31 vehicles per linear kilometer;
  - **Low Density**: 8 vehicles per linear kilometer;
- **Beacon Interval [s]**: Time interval between two consecutive transmissions of messages. In our experiments, this value is set to 1s, which is the default value of the used VANETs simulator.
- **Communication range [m]**: Communication range of nodes. Generally between 50 and 80 meters. For our simulation, this value is 70m.

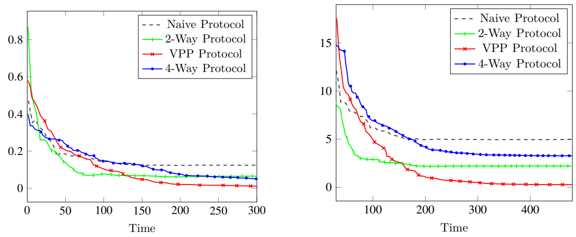
D. Experimental Results

The experimental results show the suitability of VPP for the implementation of Population Protocols in VANETs, in different scenarios.

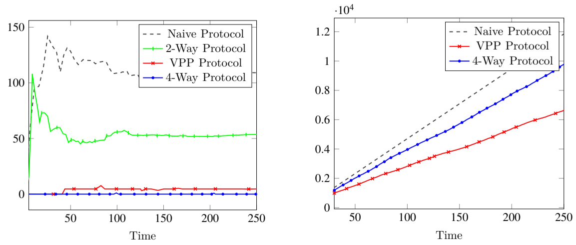
Figure 3 allows us to compare the four analyzed protocols through their Mean Absolute Percentage Error (MAPE), Root Mean Square Error (RMSE), specific error function ξ and the number of packets sent in a *Low-Density* scenario over a *Small Map*. VPP achieves the smallest MAPE and RMSE compared to both the naive protocol and to the 2-way protocol, which is an expected result, but also with respect to the 4-way protocol. This counterintuitive result is due to the excessive time span of the 4-way protocol if compared with the contact time of vehicles. Such a feature causes more inconsistent transitions than VPP. By considering the algorithm-dependent error function ξ, VPP and the 4-way protocol obtain comparable results and outperform both the naive protocol and the 2-way protocol.

Figure 4 shows the same results, obtained in the same small map, but considering a high density of vehicles. Also in this case VPP outperforms other protocols, especially with respect to the MAPE and the RMSE. We can observe comparable results also in the *Low-Density* and *High-Density* scenarios over a *Big Map*, respectively shown in Fig. 5 and Fig. 6.

Experimental results also show that VPP allows to reduce the number of transmitted packet. Figures 3d, 4d, 5d, 6d show the number of transmitted packets for the 2-way protocol, for VPP, for the naive protocol and for the 4-way protocol, in the different considered scenarios.



(a) MAPE (b) RMSE



(c) ξ (d) Number of sent packets

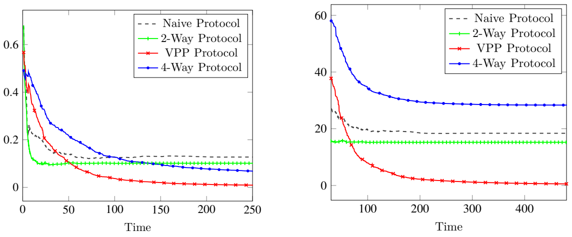
Fig. 3: Comparison of the Mean Absolute Percentage Error (MAPE) (a), of the Root Mean Square Error (RMSE) (b), of the specific error function ξ (c) and of the number of packets sent (d), for the four analyzed protocols, in a *Low-Density* scenario over a *Small Map*.

The lower number of packets transmitted by VPP compared to the 4-way protocol is an expected result. It is worth noticing that the naive protocol and the 2-Way protocol (which is not shown as its performance is perfectly comparable to the naive protocol) do not have any mechanism that stops the broadcasting phase. This mechanism is instead included in VPP and in the 4-way protocol. So the first pair of protocols performs worse than the second.

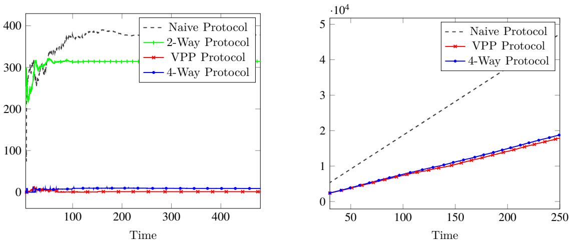
The obtained results thus show that, in three out of four scenarios, VPP achieves the best results both as regards the committed error and the number of transmitted messages. In the one scenario where the 4-way protocol produces fewer messages, its MAPE and RMSE performance is significantly worse than VVP, so VPP is still largely preferable.

VI. CONCLUSIONS

We studied the possibility of adopting the Population Protocols model in a real-world scenario where the theoretical assumptions required for its convergence are not guaranteed. Specifically, we considered the application domain of VANETs, which is characterized by dynamic network topologies and in which wireless communications can be subject to interferences and transmission errors. We analyzed the behavior of VANET nodes and identified two main problems to be addressed to achieve our goal. First, the consistency of interactions between a pair of nodes is not guaranteed due to node mobility and to corruption and loss of packets. Moreover,

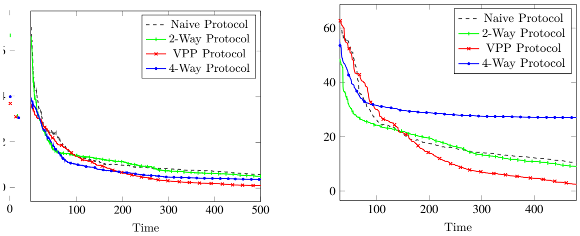


(a) MAPE (b) RMSE

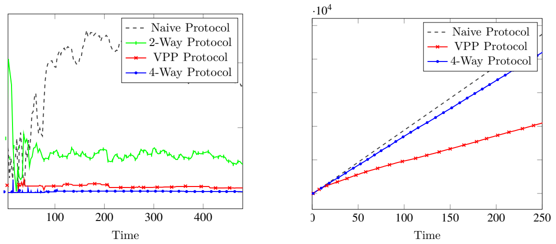


(c) ξ (d) Number of sent packets

Fig. 4: Comparison of the Mean Absolute Percentage Error (MAPE) (a), of the Root Mean Square Error (RMSE) (b), of the specific error function ξ (c) and of the number of packets sent (d), for the four analyzed protocols, in a *High-Density* scenario over a *Small Map*.

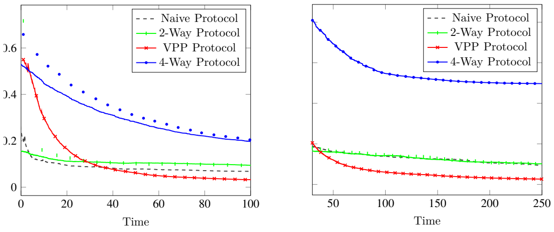


(a) MAPE (b) RMSE

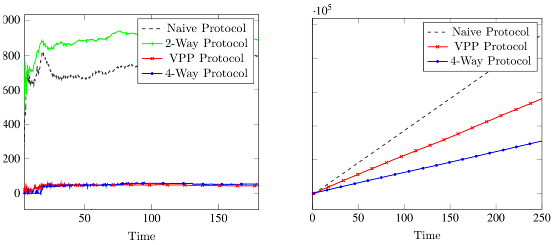


(c) ξ (d) Number of sent packets

Fig. 5: Comparison of the Mean Absolute Percentage Error (MAPE) (a), of the Root Mean Square Error (RMSE) (b), of the specific error function ξ (c) and of the number of packets sent (d), for the four analyzed protocols, in a *Low-Density* scenario over a *Big Map*.



(a) MAPE (b) RMSE



(c) ξ (d) Number of sent packets

Fig. 6: Comparison of the Mean Absolute Percentage Error (MAPE) (a), of the Root Mean Square Error (RMSE) (b), of the specific error function ξ (c) and of the number of packets sent (d), for the four analyzed protocols, in a *High-Density* scenario over a *Big Map*.

nodes must be able to distinguish their roles during an interaction, to support Population Protocols with asymmetric state update functions, without introducing global node identifiers.

To solve these issues we proposed a three-way communication schema, named Population Protocol for VANET (VPP), which uses acknowledgements and timers to obtain an adequate level of reliability of interactions before performing state updates. Moreover, the adoption of dynamic and random identifiers allows the establishment of different roles during a single interaction.

We validated our approach by considering a Population Protocol defined to solve the Counting Problem as a case study. The experimental evaluation was performed by simulating different scenarios and by comparing our solution with other possible communication schemes. The obtained results show that VPP represents the best solution to obtain good accuracy while maintaining a low communication overhead.

REFERENCES

[1] O. S. Al-Heety, Z. Zakaria, M. Ismail, M. M. Shakir, S. Alani, and H. Alsariera, "A comprehensive survey: Benefits, services, recent works, challenges, security, and use cases for sdn-vanet," *IEEE Access*, vol. 8, pp. 91 028–91 047, 2020.

[3] M. El Zorkany, A. Yasser, and A. I. Galal, "Vehicle to vehicle "v2v" communication: Scope, importance, challenges, research directions and future," *The Open Transportation Journal*, vol. 14, no. 1, 2020.

[2] S. Gaglio, G. Lo Re, G. Martorella, and D. Peri, "A middleware to develop and test vehicular sensor network applications," in *2019 AEIT International Conference of Electrical and Electronic Technologies for Automotive, AEIT AUTOMOTIVE 2019*, 2019.

[4] Y. Lin, P. Wang, and M. Ma, "Intelligent transportation system (its): Concept, challenge and opportunity," in *2017 IEEE 3rd International Conference on Big Data Security on Cloud (bigdatasecurity), IEEE International Conference on High Performance and Smart Computing (hpsc), and IEEE International Conference on Intelligent Data and Security (ids)*. IEEE, 2017, pp. 167–172.

[5] M. M. Hamdi, O. A. R. Al-Dosary, O. A. S. Alrawi, A. S. Mustafa, M. S. Abood, and M. S. Noori, "An overview of challenges for data dissemination and routing protocols in vanets," in *2021 3rd International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)*. IEEE, 2021, pp. 1–6.

[6] F. Belamri, S. Boulfekhar, and D. Aissani, "A survey on qos routing protocols in vehicular ad hoc network (vanet)," *Telecommunication Systems*, pp. 1–37, 2021.

[7] 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.

[8] 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. [Online]. Available: https://doi.org/10.1145/3293611.3331631

[9] 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.

[10] D. Doty and D. Soloveichik, "Stable leader election in population protocols requires linear time," *Distributed Computing*, vol. 31, no. 4, pp. 257–271, 2018.

[11] R. Guerraoui and E. Ruppert, "Even small birds are unique: Population protocols with identifiers," Tech. Rep., 2007.

[12] O. Michail, I. Chatzigiannakis, and P. G. Spirakis, "Mediated population protocols," *Theoretical Computer Science*, vol. 412, no. 22, pp. 2434–2450, 2011.

[13] 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.

[14] 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.

[15] J. Aspnes and E. Ruppert, "An introduction to population protocols," *Middleware for Network Eccentric and Mobile Applications*, pp. 97–120, 2009.

[16] S. Gaglio, G. Re, G. Martorella, and D. Peri, "Dc4cd: A platform for distributed computing on constrained devices," *ACM Transactions on Embedded Computing Systems*, vol. 17, no. 1, 2017.

[17] A. Bordonaro, S. Gaglio, G. Lo Re, G. Martorella, and D. Peri, "Onboard energy consumption assessment for symbolic execution models on embedded devices," in *IEEE Symposium on Emerging Technologies and Factory Automation, ETFA*, 2020, pp. 1359–1362.

[18] A. Mahmood and R. Jäntti, "Packet error rate analysis of uncoded schemes in block-fading channels using extreme value theory," *IEEE Communications Letters*, vol. 21, no. 1, pp. 208–211, 2016.

[19] 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, pp. 35–42.