



HAL
open science

Adaptive data collection protocol using reinforcement learning for VANETs

Ahmed Soua, Hossam Afifi

► **To cite this version:**

Ahmed Soua, Hossam Afifi. Adaptive data collection protocol using reinforcement learning for VANETs. IWCMC 2013: The 9th IEEE International Wireless Communications and Mobile Computing Conference, Jul 2013, Cagliari, Italy. pp.1040 - 1045, 10.1109/IWCMC.2013.6583700 . hal-00865289

HAL Id: hal-00865289

<https://hal.science/hal-00865289>

Submitted on 7 Oct 2013

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.

Adaptive Data Collection Protocol using Reinforcement Learning for VANETs

Ahmed Soua, and Hossam Afifi

Institut Mines-Telecom, Telecom SudParis, France

Email: firstname.name@telecom-sudparis.eu

Abstract—Data Collection is considered as an inherent challenging problem to Vehicular Ad-Hoc networks. Here, an Adaptive Data cCollection Protocol using rEinforcement Learning (ADOPEL) is proposed for VANETs. It is based on a distributed Qlearning technique making the collecting operation more reactive to nodes mobility and topology changes. A reward function is provided and defined to take into account the delay and the number of aggregatable packets. Simulations results confirm the efficiency of our technique compared to a non-learning version and demonstrate the trade-off achieved between delay and collection ratio.

Keywords—Data collection; Vehicular Ad Hoc Networks (VANETs); Reinforcement learning; Qlearning; Collection ratio; Number of hops.

I. INTRODUCTION

Most VANET applications are based on a dissemination process [1] [2] [3] [4] on which an information must be propagated to rather long distance so that drivers can be alerted in advance. Since each vehicle in a vehicular environment can detect a hazardous situation or a congestion zone, the number of messages pumped on the network might increase dramatically. Consequently, the network performances are severally affected leading to bandwidth waste, large overhead and a high probability of wireless collision. Thus, data gathering/collection is regarded as an important approach to circumvent these problems. It makes inter-vehicle communications more efficient and reliable and minimizes the bandwidth utilization.

In this paper, we propose an adaptive data collection scheme for VANET, namely ADOPEL. Our proposal deals with non sensitive delay applications such as traffic control applications (traffic estimation, average speed, etc).

The basic idea of the ADOPEL scheme is to choose the relay node that offers a better chance to collect more data without losing the way to the destination node. This choice is also based on the relay node's experience acquired during its passing through the network.

For that purpose, our data collection proposal is based on a distributed learning approach. A reward function for the learning algorithm is proposed to take into account the aggregation efficiency and the end-to-end delay in the collect operation.

The reminder of this paper is structured as follows. Section II explores the most relevant strategies for data collection in the context of VANETs and details the motivation of this paper. Section III introduces the proposed technique and presents the functionalities. Section IV is reserved to the validation of our

proposed technique by simulations. A conclusion follows in last section.

II. RELATED WORK

Data aggregation related literature reveals two main aspect of solving data aggregation issues. On one hand, some works focus on the manner of routing the aggregatable messages along farther distance in order to improve the aggregation ratio (data collection). On the other hand, other studies concentrate on expressing data to be aggregated differently by using compressing and merging methods to reduce the overhead. In our case, we focus on how to route the aggregatable packets to a specific destination node in order to improve the data collection ratio and hence have more accurate global traffic information.

Several works have been already accomplished to investigate the data collection concept by adopting different approaches. Thus, several studies worth our special mention here.

In [5], authors propose a location service management protocol that solves the location querying and updating problems by aggregating the location information data. In this scheme, the vehicle's mobility space is viewed as a grid network which is partitioned into several segments and each segment is divided into a number of cells. The central node of a segment plays the location server role. This server is responsible for storing current location information about all nodes belonging to the same segment. Then, the server aggregates this information and broadcast it to the neighbors. In addition, the protocol uses message aggregation in location querying. It introduces some delays before forwarding the queries in order to gather more queries and aggregate them.

This proposal is based on poor flooding to disseminate data on the network which presents a great weakness for this approach. In addition, the choice of the grid structure for the vehicle's mobility space is not justified and makes some ambiguous in this work.

B. Yu et al. in [6] focus on making similar reports broadcasted by vehicles meeting each other in order to be aggregated together. In fact, this technique dynamically changes the forwarding speed of nearby reports so that they can be delivered to the same node at the same time and then be merged into a single report. This adaptive forwarding is based on a distributed learning algorithm on which each node learns from local observations and chooses a delay based on the learning results. Simulation results outline the effectiveness of the proposed technique.

In [7], authors present their proposal, called CASCADE, where they expose a new clustering-based data aggregation technique. This protocol uses two types of reports: primary and aggregated records. The first ones are broadcasted periodically by the nodes and comprise the local view of each vehicle. Then, the local view is grouped into clusters and used to compact and aggregate the local view data into an aggregated record. This aggregated record is then broadcast to neighboring vehicles to provide them information about vehicles beyond the local view. This technique allows vehicles to have an extended view of the road behind and then accurate information about upcoming traffic conditions.

Another effort carried out by Y. Dieudonne et al. in [8] focuses on a distributed collection information for VANETs. It collects data produced by vehicles using inter-vehicle communications only. It is based on the operator *ant* allowing to construct a local view of the network and therefore to collect data in spite of the network topology changes. A theoretical proof of correctness and experiments confirm the efficiency of the proposed technique.

Nadeem et al.[9] introduce a system for data dissemination and aggregation in a vehicular context namely Traffic View. In this system, an aggregate record is composed of specific information: single speed, position, timestamp value and a list of vehicle's IDs. The authors propose two aggregation schemes: ratio and cost based techniques. In the ratio-based, the most important parameter is the aggregation ratio which indicates the number of vehicles to be aggregated into one single frame. For the cost based technique, a specific cost function is defined for each aggregating vehicle. A high cost is assigned for the vehicles that are close to the aggregating node. Thus, the produced view of the traffic is not useful to any vehicle unless it is in the proximity of the aggregating vehicle.

Lochert et al. in [10] [11] focus on cooperative information gathering and sharing applications in VANETs and propose a hierarchical aggregation algorithm. Their proposal is based on a probabilistic data representation Flajolet-Martin sketches, which they extended to a soft-state data structure. In their scheme, there is no longer a need to decide which aggregate contained more up-to-date information since the resulting aggregate comprises all the information from all aggregates that have been merged. Nevertheless, this work does not consider routing related-issues but focus only on data representation.

The aforementioned aggregation/collect approaches do not strictly consider the potentially mobility issue and the collection ratio in finding a suitable relay in the collect process. In fact, most of the listed works focus on the representation and the processing of the aggregated data and neglect how to obtain the raw information among the running vehicles. Thus, our proposal is interested on collecting aggregatable packets from vehicles taking into account the mobility of vehicles and the dynamicity of the network. We use Q-learning method to select next hops aiming at collecting more raw data.

III. PROPOSED TECHNIQUE

In this section, we introduce the ADOPEL protocol- an adaptive distributed data collection scheme for VANETs. The

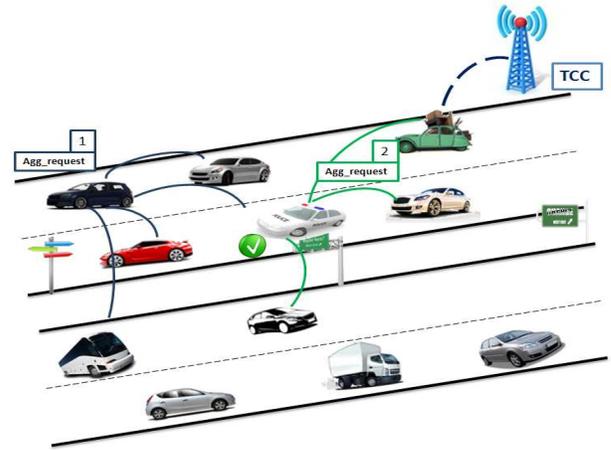


Fig. 1: ADOPEL functioning Overview

proposal is based on a distributed learning algorithm on which a reward function is defined. This latter takes into account the delay and the number of aggregatable packets and hence makes the collection operation more reactive to nodes mobility and topology changes. Thus, our technique aims at maximizing the collect ratio and respecting the end to end delay required by the application.

A. System Specifications

ADOPEL considers that each communicating vehicle knows its current position and speed using a positioning system such a Global Positioning System (GPS). Furthermore, we assume that vehicles exchange two types of messages: beacons and event driven-messages. When the former aims at improving driver awareness of surrounding environment by exchanging information about position, velocity, direction, etc., the latter is triggered when a vehicle needs to collect traffic data toward a control center.

Our scheme deals with providing data collection service for non-sensitive delay applications such as congestion detection warning, available parking spaces, etc.

This collect operation is started by a node called *initiator* and involves a limited number of vehicles. Here, the *initiator* is a vehicle that is leading a group of nodes and running in a highway. The *initiator*, at each gathering operation, is randomly selected from vehicles.

Thereby, the *initiator* have to collect the traffic data from vehicles and deliver it to a Traffic Control Center (*TCC*) in order to be processed and studied (see Figure 1). We assume here that *TCCs* are sufficiently deployed along the freeway.

In a vehicular context, the collect of traffic related data is periodically carried out and transmitted to a *TCC* in order to have an up to date big picture of the road. Thus, ADOPEL triggers periodically a collect operation from the *initiator* toward a *TCC*. The collected data is provided to the *TCC* when this latter is reachable by the node ending the gathering operation. To limit the collection process, we use a $d_{collect}$ parameter, representing the depth of the collecting operation, i.e. the maximal distance in meters from the initiator. Indeed,

this parameter reflects the zone that will be concerned by the collecting process. Thus, each additional meter increases the total duration of the collect as well as the number of messages to collect; $d_{collect}$ is then an interesting parameter, impacting directly the performances of our proposal. The type of data to be collected is specified by the *initiator* and included in the collect packets. For instance, in our scenarios, ADOPEL deals with collecting the speed of surrounding vehicles with the aim of computing the average speed of the concerned road. However, this data type can be extended to other useful information as well as real-time fuel consumption, pollution indicators and parking lots availability services, etc. We have to mention here, that we are not interested on how to aggregate data and express it differently. As mentioned previously, we focus on the manner of routing the aggregatable messages (selecting the appropriate relay) along farther distance in order to improve the data collection ratio.

B. ADOPEL protocol design

1) Distributed Qlearning in ADOPEL:

The frequent topology changes in the vehicular context make it necessary to adapt the aggregation and the forwarding policy to the network state. In fact, it is difficult to predict in advance the set of rules that will adjust the actions of each vehicle when the vehicular environment's variables are changing.

Fortunately, the reinforcement learning techniques [12] can tackle these problems. In reinforcement learning, each vehicle is a learner. Each vehicle tries to optimize its interactions with the very dynamic environment through its experience. The experience here is expressed in terms of errors and rewards. In addition, the vehicles collaborate with each other to share their feedbacks and establish the distributed learning system.

In this work, we model the aggregation operation in VANET as a Markov decision problem which can be solved by reinforcement learning. Each vehicle (agent) decides at each state which action to take based on its experience. After taking an action, the agent gets a reward or a cost from the environment. The Markov decision problem is defined as a tuple $\{s, a, r\}$

- s is the states set; In our work, the packet state is the current vehicle.
- a is the set of actions a vehicle can perform: in our scheme, the action of a node is to select the next relay that will maximize the aggregation ratio. Hence, the possible set of actions allowed at each node is nothing but the set of neighbors.
- r is the immediate reward a vehicle may receive after taking an action a

To solve this MDP model, we propose to use a reinforcement learning algorithm. The literature provides a large number of reinforcement learning approaches, such as temporal difference learning, direct utility estimation and Q-learning [13]. We are motivated to use Q-learning algorithm since it allows comparing the expected utility of the available actions without requiring knowledge of the environment's model.

A $Q(s, a)$ matrix is used to store the learned reward/cost for each state and action pair. For example $Q(s, a)$ is the expected

reward for taking an action a at state s . The updating function of $Q(s, a)$ is defined as

$$Q(s, a) = (1 - \alpha) * Q(s, a) + \alpha * (r + \gamma * \max_{a'} Q(s', a')) \quad (1)$$

Where

- α denotes the learning rate which models here how quickly the Q-values can change with a dynamic network topology.
- γ refers to the discount factor. It models the fact that immediate reward is or not more valuable than future reward. If this value is high, future rewards are more valued than immediate rewards. In the opposite case, the learning algorithm count immediate rewards more strongly.
- r represents the expected immediate reward of executing action a at state s .
- $\max_{a'} Q(s', a')$ models the maximum expected future reward when the system reaches the state s' after taking the action a .

The most important challenge to successfully achieve the aggregation performance is to define the suitable reward function. In fact, the vehicle will use this function to update its forwarding policy.

For immediate rewards, we consider the most relevant parameters effective in decision. First factor is based on the number of neighboring vehicles that each node possesses in its transmission range. In fact, the reward should be more for a vehicle with a high number of neighbors. Secondly, the aggregation proposal must route the packet to the destination in a limited delay. Thus, the node has to choose the node that offers the most relevant advance to the destination. It is worth saying that our proposal focuses on a collection process rather than a rapid propagation of a packet in the network. This observation has to be considered on the reward function.

Based on these decision factors, we formulate the reward function as follows :

$$r = \begin{cases} \beta * (1 - \frac{1}{nb_neighbor(i)}) + (1 - \beta) * (\frac{adv(ij)}{adv(i)_{avg}}) & \text{if next hop is the destination} \\ k_1 & \text{if next hop is the destination} \\ -k_1 & \text{if the vehicle hasn't neighbors} \end{cases} \quad (2)$$

The reward function considers several routing scenarios to improve the aggregation ratio and guarantee a steady advance to the destination.

The first item in Eq.2 combines the normalized number of neighboring nodes that the next hop possesses and its normalized progress toward the destination. adv refers to the advance of the node i (current node) to the destination vehicle D , situated at a distance $d_{collect}$ from the *initiator*, by choosing the neighboring node j as the next hop. This parameter d can be seen as the depth in distance of the aggregation process.

Hence, this advance can be expressed as follows: $adv(ij) = dist(i, d) - dist(j, d)$. The average advance is given by: $adv(i)_{avg} = (\sum_{k=1}^t adv(ik)) / nb_neighbor(i)$ where $t = nb_neighbor(i)$ is here the total number of neighbors of node i .

Thus, more reward is assigned to the next hop with more neighbors and larger relative advance. In fact, a node with a higher number of surrounding vehicles and a higher advance toward the *TCC* allows respectively a larger quantity of collected data and a faster delay to reach the destination.

The second item in Eq.2 denotes the reward if the node can reach directly the destination *D*. In this case, the reward is a positive constant k_1 .

Finally, the last item is to solve the "void" problem in geographic routing. In fact, when a node receives a packet and cannot find a neighboring vehicle, it drops the packet and sends a negative reward to the sending node to inform a forwarding failure. Then, the sending node will choose another vehicle to send the packet based on the Q-values. The node with the highest Q-value will be selected.

As an important feature in our proposal, we use a variable discount factor called γ' to handle the instability of the vicinity. This parameter depends on the link stability. In fact, the node selected as a next relay is the vehicle that will spend more time in the vicinity of the sending vehicle. In this way, we ensure that the route we select is more stable. For that purpose, we define a stability factor SF_i as

$$SF_i = \begin{cases} \sqrt{\frac{N_i \cap N_{i+1}}{N_i}} & \text{if } N_i \neq 0 \\ 0 & \text{Otherwise} \end{cases} \quad (3)$$

Where N_i (N_{i+1} respectively) is the current neighbor set of the sending node i (the forwarding node $i + 1$ respectively). Neighbor list can be attached to the hello messages exchanged between vehicles. As aforementioned, the SF will reflect a higher value for a relatively stable couple of neighbors. Then, a node calculates the discount factor γ' as :

$$\gamma' = \gamma * \sqrt{SF_i} \quad (4)$$

Therefore, every time a node has a packet to send, it calculates the reward for its neighboring set, the stability factor and updates the Q-values of its matrix using the following equation

$$Q(s, a) = (1 - \alpha) * Q(s, a) + \alpha * (r + \gamma' * \max_{a'} Q(s', a')) \quad (5)$$

The vehicle with the highest Q-value will be selected as next hop.

2) Exploration vs Exploitation:

In reinforcement learning there is a balance between exploitation and exploration. Exploitation occurs when the action selection strategy is based on the highest value of the Q-table. In this case, exploitation will lead to locally optimal policies since the selection is greedy. In the case of most of the optimization problems, this will not lead necessary to a global optimum.

On the other hand, exploration consists on taking risk by choosing the nonoptimal action and exploring other choices to obtain more knowledge about the network. Obviously, excessive exploration degrades the performance of the Q-learning approach.

Thus, convergence is an important issue for our proposed algorithm. Nevertheless, in [14], authors demonstrate that a Q-learning scheme converges to the optimum actions-values provided that "all actions are repeatedly sampled in all states and action-values are represented discretely. Here, the conditions of convergence are insured. In fact, ADOPEL uses hello messages to sample all its neighbors by computing the γ' factor. In addition, the action-values (Q-values) are represented discretely in ADOPEL. As a result of that, we can say loudly that our proposed technique converges to the optimum action-values.

3) ADOPEL Algorithm Overview:

Based on the description given in the previous section, we summarize here the different steps of ADOPEL.

As stated above, each node uses the received "hello" messages from neighbors to build a neighboring node table. The "hello" messages contains in addition to the usual information the list of neighboring nodes. This way, each vehicle can maintain its two-hop neighbor list and can easily compute the stability factor given by Eq.3.

Algorithm Alg.1 shows the different steps of the execution of ADOPEL on a each node i whenever this latter receives a collect request. This execution is triggered periodically by an *initiator* node.

Upon receiving a relaying request, the first step undertaken by a node i aims at collectiong data from neighbors by sending them a collect data request. Afterward, the node processes the data received (eg: it computes the average value of the received ones) and starts the relaying process.

For the relaying process, it classifys the neighboring nodes on three different lists. Highest priority is attributed to vehicles that are more surrounded and closest to the final destination node situated at a distance $d_{collect}$. Notice that a vehicle with a large number of neighboring nodes leads to a larger quantity of collected data.

The second phase consists on selecting the appropriate relay node based on the previous classification. This operation depends on the Q-values of each candidate node. In fact, nodes with high values of Q are prosperous.

Once the selection of the relay vehicle is performed, the sending node computes the immediate reward r and then calculate the total expected reward $Q(s, a)$.

Since the collection process is periodically initiated by the *initiator*, a node i is involved in this operation for few times before leaving the concerned road. Thus, the vehicle learns from its acquired experience (rewards or costs) to select the appropriate relay node ensuring a good collection ratio and a faster propagation toward the destination.

IV. RESULTS

In this section, we show our simulations results and investigate the performances of our proposal in terms of collection ratio and number of hops. We compare our scheme to a non-learning protocol. We call a collection technique "non-learning" when a first part of relays are selected based on the number of their neighboring cars and the other part are selected based on their advance toward the final destination.

Algorithm 1 ADOPEL algorithm, for evry node i

```
1: Send a data collection request to neighboring nodes.
2: L1, L2 and L3 are 3 lists initialized by NULL.
3:  $Q_i$  is initialized based on the number of surrounding vehicles
   and the advance toward the destination
4:  $N_i$  is the neighboring nodes set of node  $i$ 
5: if " $N_i \neq \emptyset$ " then
6:   for  $j \in N_i$  do
7:      $i$  compares each of its neighboring node  $j$  as follows :
8:     if " $nb\_neighbor(j) \neq 0$  and ( $advance(j) > advance(i)$ )"
       then
9:        $L1 \leftarrow j$ 
10:    end if
11:    if " $nb\_neighbor(j) = 0$  and ( $advance(j) > advance(i)$ )"
      then
12:       $L2 \leftarrow j$ 
13:    end if
14:    if " $nb\_neighbor(j) \neq 0$  and ( $advance(j) \leq advance(i)$ )"
      then
15:       $L3 \leftarrow j$ 
16:    end if
17:  end for
18:  if " $L1 \neq \emptyset$ " then
19:    Next hop will be the one with largest Q-Value
20:  end if
21:  if " $L1 = \emptyset$  &  $L2 \neq \emptyset$ " then
22:    The node with largest Q-Value will be next hop
23:  end if
24:  if " $L1, L2 = \emptyset$  &  $L3 \neq \emptyset$ " then
25:    Next Hop will be chosen from L3 with highest Q-Value
26:  end if
27:   $i$  relays the message to the selected next hop after making
   aggregation (computing average value)
28: else
29:   /*  $N_i$  is empty */
30:    $i$  generates a negative reward
31:    $i$  chooses its previous source as the next hop
32: end if
33:  $i$  computes the reward after making the relaying process based
   on 2
34:  $i$  updates the Q-Value  $Q(s, a)$  using Eq.5
```

The destination is situated at a distance $d_{collect}$ behind the *initiator*.

A. Simulation Design

We used MATLAB to conduct simulations using Freeway mobility model. The freeway mobility model emulates the motion behavior of vehicles in a freeway. In our study, we use a freeway which has two lanes in each direction. All lanes of the freeway are 20 Km in length.

To make the proposed scheme tractable, we make the following assumptions:

- We assume an ideal MAC layer without contention and collision.
- All nodes have the initial transmission range equal to 200 m.
- The number of vehicles was varied from from 200 to 400.
- All vehicles are initially positioned at the entrance of the freeway.
- we respectively assigned to α , β and γ the following values: 0.8, 0.7 and 0.8.

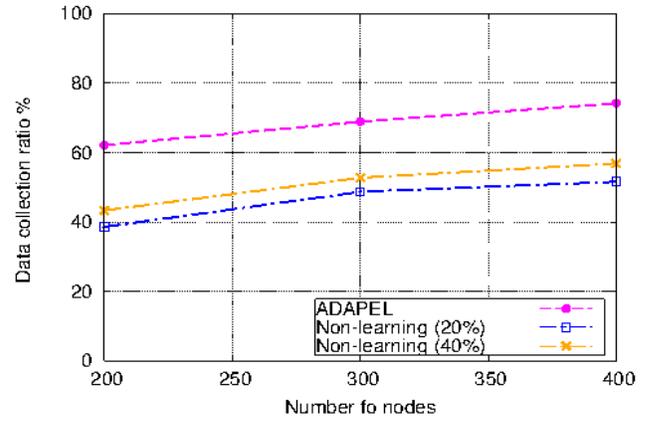


Fig. 2: The average data collect ratio: ADOPEL vs Non-learning versions

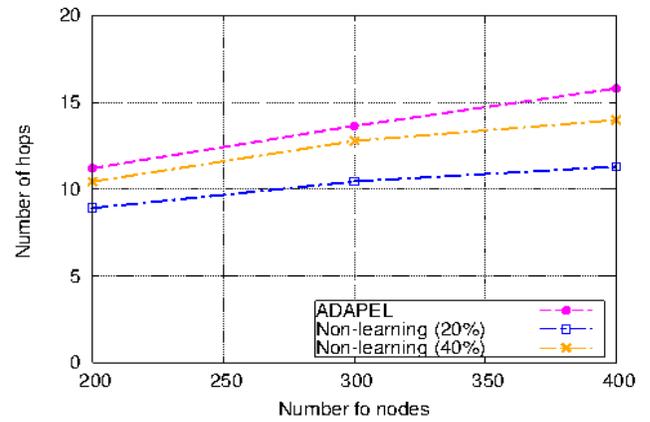


Fig. 3: The average number of hops: ADOPEL vs Non-learning versions

- For the data collection depth, we set $d_{collect}$ equal to 1500 m.

In addition, each vehicle stores its own Q-values and the ones received from neighboring nodes (using hello messages) in a matrix to be used in the relaying process.

Furthermore, we compare our scheme to a non-learning version. For that purpose, we suppose that, at each relaying operation, a node has respectively a 20 percent (40 percent) probability of choosing the most surrounding vehicles as a next relay and 80 percent (60 percent) to choose the node with the largest advance toward the destination node.

B. Simulation Results

In this section, we focus on the performances of our technique both for the average data collect ratio and the average number of hops required to reach the final destination node.

Figure 2 depicts the average data collection ratio when varying the density of nodes for the two techniques. We can observe that our proposed scheme outperforms loudly the non-learning versions. In fact, in all cases, ADOPEL achieves a gain of over than 20% compared to the other techniques. This

can be explained by the fact that in very dynamically changing networks as VANETs, ADOPEL can change adaptively to better relaying nodes to increase the collect ratio as the network topology changes, whereas the others non-learning protocols find major difficulties to adapt to the dynamicity of the network.

To make a fair analysis, we investigate in Figure 3 the average number of hops needed to travel the collect distance $d_{collect}$. Indeed, a good collect ratio might have a heavy cost and then can be a real weakness for the algorithm. However, Fig.3 shows that the gap between the three techniques is very tight, even ADOPEL achieves higher values than the others schemes. This clearly implies that our technique achieves a good trade-off between delays and collection ratio. This is because ADOPEL takes the stability of vicinity into account which yields in a higher probability of using nodes moving in the same direction as the destination node to relay aggregated messages. On the other side, in non-learning versions, the source node may select a node moving in opposite direction as a next hop which can very vulnerable. As a result, many data collection operations may be penalized when relaying vehicles became far away from the destination.

V. CONCLUSION

In this paper, we have introduced the ADOPEL protocol, a new data collection algorithm devoted to vehicular networks. ADOPEL is based on a distributed learning approach where vehicles adaptively choose the forwarding relays to maximize the ratio of collected data without losing the way to the destination node. Our simulation results corroborate the effectiveness of our scheme compared to non-learning versions.

REFERENCES

- [1] G. Badawy, J. Mistic, T.D. Todd, and D. Zhao, *Performance modeling of safety message delivery in vehicular ad hoc networks*, IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), Oct. 2010.
- [2] Y. Bi, L.X. Cai, X. Shen, and H. Zhao, *A Cross Layer Broadcast Protocol for Multihop Emergency Message Dissemination in Inter-Vehicle Communication*, IEEE International Conference on Communications (ICC), May. 2010.
- [3] R.Singh, and A.Gupta, *Information Dissemination in Vanets using zone based forwarding*, IEEE/IFIP Wireless Days Conference, Oct. 2011.
- [4] L. Chou and Y. Yang, *Location-Based Directional Broadcast for inter-vehicle Communications*, IEEE Vehicular Technology conference, Sept. 2010.
- [5] H. Saleet, O. Basir, R. Langar, and R. Boutaba, *Region-Based Location-Service-Management Protocol for VANETs*, IEEE Transactions on Vehicular Technology, vol. 59, no. 2, FEB. 2010.
- [6] B. Yu, C. Xu, and M. Guo, *Adaptive Forwarding Delay Control for VANET Data aggregation*, IEEE Transactions on Parallel and Distributed Systems, vol. 23, no. 1, pp. 11-18, Jan. 2012.
- [7] K. Ibrahim, and M. Weigle, *CASCADE: Cluster-based Accurate Syntactic Compression of Aggregated Data in VANETs*, GLOBECOM Workshops, Nov. 2008.
- [8] Y. Dieudonn, B.Ducourthial, and S.M. Senouci, *COL: a Data Collection Protocol for VANET*, Intelligent Vehicles Symposium, Jun. 2012.
- [9] T. Nadeem, S. Dashtinezhad, C. Liao, and L. Iftode, *TrafficView: Traffic Data Dissemination using Car-to-Car Communication*, ACM Mobile Computing and Communications Review (MC2R), vol. 8, no. 3, pp. 6-19, Jul.2004.
- [10] C. Lochert, B. Scheuermann, M. Mauve. , *Probabilistic Aggregation for Data Dissemination in VANETs*, ACM International Workshop on Vehicular Ad Hoc Networks, pp. 1-8, Sept. 2007.
- [11] C. Lochert, B. Scheuermann, M. Mauve, *A Probabilistic Method for Cooperative Hierarchical Aggregation of Data in VANETs*, Ad Hoc Networks, vol. 8, no. 5, pp. 518-530, Jul. 2010.
- [12] L. Panait, and S.Luke, *Cooperative Multi-Agent Learning: The State of the Art*, Autonomous Agents and Multi-Agent Systems, vol. 11, no. 3, pp. 387-434, Nov.2005.
- [13] S.J. Russel, and P. Norvig, *Artificial Intelligence: A Modern Approach*, second ed. Prentice Hall, 2003.
- [14] C.J.C.H, Watkins, and P. Dayan, *Q-learning*, Mach. Learn, vol. 8, no. 3-4, pp. 279-292, 1992.