Decentralized Cooperative Piggybacking for Reliable Broadcast in VANET

Guangbing Xiao, Haibo Zhang, Zhiyi Huang, and Yawen Chen
Department of Computer Science, University of Otago, New Zealand

***Abstract*—Reliably broadcasting safety information to neighboring vehicles is a big challenge in vehicular ad-hoc networks (VANETs), due to the dynamic network topology and the unreliable wireless channels. In this paper we present two decentralized cooperative schemes to enhance broadcast reliability by exploiting the advantage of message piggybacking. The key idea is to let each vehicle optimally piggyback some messages it has received when broadcasting with the expectation that the neighboring vehicles can recover its lost messages through the piggybacked messages. We first present greedy piggybacking, in which each vehicle announces its lost messages to neighboring vehicles and makes piggybacking decisions based on message losses in its neighbors. We observed that some lost messages still cannot be successfully recovered in greedy piggybacking due to the asymmetric wireless communications, and further proposed a mutual learning based scheme to overcome the drawback of greedy piggybacking. We evaluated the performance of the two schemes through trace-driven simulations, and results show that both schemes can achieve significant improvement on broadcast reliability in VANETs in comparison with the existing solutions.**

***Index Terms*—Cooperative Broadcast, Piggybacking, Broadcast Island, Reliability.**

I. Introduction

The rapidly growing demand for the safe and comfortable *smart driving* makes the reliable wireless connectivity a critical factor in the vehicular ad-hoc network(VANET).

Existing solutions to solve the reliable broadcast problem can be broadly grouped into three categories: (1) *ACK/NACK based source retransmission* [3]: the receiving vehicles use NACK or implicit ACK to notify the sending vehicle of the message loss, and the sending vehicle will retransmit the lost message. This scheme is simple but may not be efficient as message loss is generally bursty. If the current message is lost, there is a large probability for the retransmitted message to be lost over the same channel.(2) *Flooding-based Forwarding* [7]: Any vehicle that lost the message will broadcast a retransmission request, and all other vehicles that overhear the request will help forward the lost message if they have received it. However, the vast expanse of duplications are prone to severe broadcast storms, making itself a DoS attack together with unpredictable delay. (3) *Cooperative Forwarding* [9] [10]: to avoid the broadcast storms, the forwarders are carefully selected with the most reliable links to forward the lost message. Unfortunately, seeking such an optimal forwarder is not always an easy task due to the time-varying network topology and wireless channel quality.

We aim to develop efficient schemes to enhance the reliability of CAM broadcasts by exploring the advantage of *cooperative message piggybacking*. Whenever a vehicle broadcasts a CAM, it can piggyback some CAMs it received from other vehicles. All vehicles cooperatively select some CAMs to piggyback, with the expectation that each vehicle can recover its lost CAMs from the CAMs piggybacked in other vehicles' CAMs. In comparison with existing solutions, our scheme has at least the following two advantages: (i) No ACK/NACK is used to recover the lost CAMs, thereby avoiding broadcast storms; (ii) the broadcast goodput can be improved by optimally selecting the piggybacked messages, as the overhead incurred by message headers is reduced. For example, the IEEE 1609.2 standard uses the ECDSA algorithm for authentication. The data payload that contains state information is only 53 bytes, whereas the certificate and signature used for authentication take up 209 bytes [6]. The key contributions of this work are summarized as follows:

- We proposed a greedy piggybacking scheme, in which each vehicle piggybacks the information of its lost CAMS when broadcasting its own CAM, and each vehicle makes piggybacking decisions based on the message loss information received from its neighbours.
- We observed that some lost CAMs cannot be recovered by greedy piggybacking due to the asymmetric communication links, and proposed a mutual learning scheme to solve this problem. We derived the expected converge speed for mutual learning, and gave both a lower bound and upper bound on convergence time.
- We evaluated our schemes through simulations using real V2V communication traces, and results show that our schemes can achieve significant improvement on broadcast reliability in comparison with existing solutions.

The rest of this paper is organised as follows. Section II reviews the related work on improving broadcast reliability in VANETs. Section III gives the system model and the statement of the problem. In Section IV, the greedy piggybacking scheme and the scheme based on mutual learning are presented and discussed. Section V presents the simulation results and the paper is concluded in Section VI.

II. Related Work

Many researches have been carried out to improve broadcast reliability in VANETs by solving the hidden terminal problem [4] [5]. Ma et al. proposed a dynamic scheme to disseminate emergency warning messages, by letting adjacent vehicles in different directions to forward the lost message repeatedly before the message deadline [7]. Although multiple copies from different directions can bypass signal blocks and noise in some degree, such a duplication flooding will cause severe broadcast storms. In [8] a double-covered broadcast algorithm was proposed. In this scheme the forwarding nodes are selected in such a way that 1) the sender's 2-hop neighbors are covered and 2) the sender's 1-hop neighbors are either forwarding nodes or nonforwarding nodes covered by at least two forwarding neighbors. However, the authors assumed that the receiver will receive the forwarded message as long as the sender overhears the forwarder's retransmission, which is not always true in VANETs.

Most existing cooperative forwarding approaches either require global network information to select the forwarder or select forwarders not based on the quality of the communication links. In [9] a receiver consensus based approach was proposed for forwarder selection. The vehicle nearest to the ideal forward position will be selected based on geographic information. The drawback is that each vehicle has to dynamically maintain geographic information of other vehicles in the VANET. In [10] Sharma et al. studied the resource allocation in decentralized information local public good networks and induced global solutions at Nash equilibriums via centralized allocation. However, centralized control by a network manager is very difficult and even impossible in VANET.

III. System Model and Problem Formulation

A. System Model

We assume all vehicles are equipped with radio transceivers, and different vehicles may have different radio transmission ranges. Each vehicle employs a virtual radar on the panel screen to dynamically monitor the status of neighbouring vehicles. The monitoring range for each vehicle is defined as a disk with radius of r_i centred at the current position of the vehicle. Let v_i be an arbitrate vehicle in the VANET and $N_i = \{n_{i1}, n_{i2}, \cdots, n_{in}\}$ be the set of neighbouring vehicles in v_i's monitoring range. We assume that each vehicle v_i broadcasts a CAM every δ milliseconds, and the deadline for the CAM coincides with the refresh interval. Each vehicle v_i monitors the status of all neighbouring vehicles in N_i based on the CAMs received. Any neighbouring vehicle in N_i that v_i cannot receive its CAM reliably, will disappear on v_i's virtual radar. Each vehicle v_i also reserves buffer B_i to store the CAMs received from its neighbours. For any CAM buffered in B_i, it is not eligible to be piggybacked and will be discarded if its deadline expires.

In our model, we use the random access protocol IEEE 802.11p, which is based on the Carrier Sense Multiple Access (CSMA), to access the control channel, as it has been chosen as the standard for the communications in VANETs. We assume time is synchronised with GPS, and divided into slots with equal length. As shown in Fig. 1, the slots are organised into frames, and each vehicle maintains its local frame that records the transmission schedule of its neighboring vehicles. We assume that the length of a frame is smaller than δ milliseconds so that each vehicle can be allocated a slot in the frame, and the frame is repeated for periodic broadcasting. The

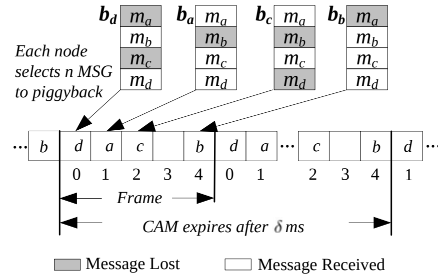

Fig. 1. Piggybacking CAMs via STDMA in the VANET

length of the frame maintained at a vehicle is proportional to the number of neighboring vehicles, and can be dynamically changed when vehicle joins or leaves its neighborhood.

B. Problem Statement

To monitor the status of neighbouring vehicles, each vehicle needs to reliably receive the CAMs broadcast by vehicles in its RoI. However, a vehicle may not receive the CAMs of another vehicle in its RoI due to the lossy and unstable wireless channel or the asymmetric wireless links. In this paper we exploit the advantages of message piggybacking to enhance the reliability of CAM broadcast.

Whenever a vehicle v_i broadcasts a CAM, we assume it can choose to piggyback at most $w(w \leq n)$ CAMs stored in its buffer B_i. Let $P_i^j = \{m_1, m_2, \cdots, m_w\}$ represent the set of w CAMs piggybacked by vehicle v_i when broadcasting its own CAM m_j. We use P^j to denote the set of piggybacking decisions for all vehicles in the network during the lifetime of message m_j. Let $\bar{n}_i^j$ represent the average number of vehicles in v_i's RoI in the lifetime of m_j, and $n_r^j(P^j)$ denote the number of vehicles that receive m_j based on the piggybacking decisions given in P^j. We define the broadcast reliability of message m_j that was generated by v_i, denoted by R_i^j, as

$$R_i^j = \frac{n_r^j(P^j)}{\bar{n}_i^j}. \tag{1}$$

We aim to maximize the average broadcast reliability that is defined as follows:

$$\frac{1}{|V|} \sum_{v_i \in V} \frac{1}{|M_i|} \sum_{m_j \in M_i} \frac{n_r^j(P^j)}{\bar{n}_i^j}, \tag{2}$$

where V is the set of vehicles in the network, and M_i is the set of CAMs broadcast by vehicle v_i. We assume that each vehicle does not have global network information such as the network connectivity, the piggybacking decisions of other vehicle as well as their real-time buffer status, etc. Each vehicle will make the piggybacking decision based on information only from local buffer and neighbours, with the expectation to converge to the global optimal solution.

IV. Cooperative Piggybacking

A. Greedy Piggybacking

If a vehicle knows which CAMs its neighbours have lost, broadcast reliability can be enhanced by letting the vehicle piggyback those lost messages when broadcasting its own CAM. The key ideas of greedy piggybacking (GP) are: (a) each vehicle piggybacks the request for its lost CAMs when broadcasting its own CAMs; (b) each vehicle makes piggybacking decisions based on the requests received from its neighbours. The more times a cached CAM was requested, the higher the priority it will be piggybacked.

Since each vehicle knows the transmission schedule of its neighbors, the failure of receiving a CAM can be easily detected at the receiving vehicles. When a vehicle v_i fails to receive a CAM broadcast by vehicle v_j, it will empty the corresponding buffer $B_i[v_j]$. Algorithm 1 gives the pseudocode to be executed at each vehicle v_i for CAM broadcast.

Algorithm 1: Greedy Piggybacking at Vehicle v_i

```
Upon broadcasting CAMs:
  for each lost CAM in B_i do
    Add the source ID of the lost CAM to srq_i;
  Choose w cached CAMs to piggyback;
  Broadcast its CAM by piggybacking srq_i and the
  selected w CAMs;
Upon receiving a CAM from v_j:
  for each item in srq_j do
    if the requested CAM is in B_i then
      Increase the request count for this CAM.
  for each item in cam_j do
    if there is no such CAM in B_i then
      store it in B_i.
    else
      keep the copy with the latest timestamp.
Upon CAM deadline expires:
  Discard the expired CAM.
```

Upon broadcasting a CAM, v_i first checks whether there is any lost CAMs, and generate a request vector $\boldsymbol{srq}_i$ that includes the information of its lost CAMs. The CAMs cached in B_i are ranked based on the number of requests received from its neighbors in a non-increasing order, and the first w CAMs are selected for piggybacking. At the end, v_i includes its own CAM together with the w selected CAMs into a vector $\boldsymbol{cam}_i$, and broadcasts it with $\boldsymbol{srq}_i$.

Upon receiving a CAM from v_j, for each item in $\boldsymbol{srq}_j$, v_i checks if it has a copy of requested CAM in its buffer. If it has a copy in its buffer, it increases the requested count for that CAM. For each CAM in $\boldsymbol{cam}_j$, it is stored in B_i if the corresponding buffer in B_i is empty. If there is already a copy in B_i, and two copies will be compared based on the timestamps, and the new one is stored.

Each CAM cached at B_i is associated with a timer, and the timeout value is set to the CAM's lifetime. If the timer expires, the corresponding CAM is deleted from B_i.

B. Mutual Learning Piggybacking

It seems that GP is efficient as each piggybacked CAM is carefully selected based on the lost requests. However, GP chooses the piggybacked CAMs too greedily, which cannot solve the following problem, denoted at the Broadcast Island Problem (BIP), that may frequently occur in VANETs.

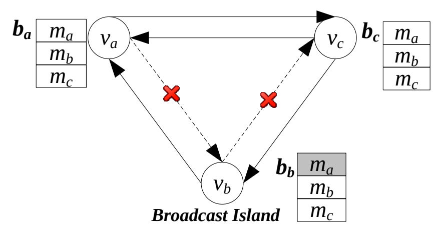

Fig. 2. Broadcast Island Problem in the VANET

As shown in Fig. 2, we assume that v_b cannot hear from v_a, and v_c cannot hear from v_b. When v_b lost the CAM m_a, it will broadcast a request for m_a according to GP. In fact, it is impossible for v_b to regain the lost m_a because: (1) v_a can hear v_b's request, but its piggybacking cannot be heard by v_b; (2) v_c's piggybacking can be heard by v_b, but v_c cannot hear v_b's request. The reasons why GP cannot solve BIP can be summarized as follows: (1) wireless links are asymmetric due to the antenna's imperfect disk propagation and the different transmission powers used by different vehicles; (2) each vehicle makes its piggybacking decisions merely based on its local incomplete information. In the following, we present a mutual learning based solution to solve this problem.

The key idea of mutual learning based piggybacking is to let each vehicle give suggestions to neighboring vehicles on which CAMs it expects them to broadcast, and each vehicle piggybacks the CAMs according to the received suggestions from its neighbours. We use g_{ij} to indicate the suggestion from v_i to v_j on which CAMs to piggyback, and the combination of suggestions from v_i to all other vehicles, including v_i itself, is defined as v_i's proposal $\boldsymbol{g}_i = \{g_{i1}, g_{i2} \cdots g_{im}\}$. Each vehicle v_i piggybacks its proposal $\boldsymbol{g}_i$ when broadcasting its own CAMs, and v_i makes its own suggestion to each v_j based on its neighbours' suggestions to v_j in the received proposals. The CAM with the max suggested times to v_j in the received proposals will be suggested again by v_i. All the suggestions from v_i to each v_j is constructed as v_i's proposal $\boldsymbol{g}_i$. Similar to GP, each vehicle makes piggybacking decisions according to the suggestions in the received proposals. The more times a CAM was suggested by the neighbours, the higher the priority it will be piggybacked. Algorithm 2 gives the pseudocode to be executed at each v_i for CAM broadcast.

We use the example given in Fig. 2 to illustrate how MLP can solve BIP. When v_b lost m_a, it will allocate the lost CAMs in its proposal, say $\boldsymbol{g}_b = \{g_{ba} = m_a, g_{bb} = m_b, g_{bc} = m_a\}$. When v_a receives this proposal, it will combine this information in its proposal, say $\boldsymbol{g}_a = \{g_{aa} = m_a, g_{ab} = m_b, g_{ac} = m_a\}$, and piggybacks m_a. As v_b cannot hear v_a's piggybacking, the lost CAM m_a still cannot be regained at this broadcast. However, v_c can hear v_a's proposal, and it will realise that someone is requiring for m_a. Hence, it accommodates this suggestion in its own

Algorithm 2: Mutual Learning Piggybacking Algorithm

```
Upon broadcasting CAMs
```

for *each row j in* $\mathbf{rs}_i$ **do**

  Rank the suggestion to v_j;

  Set the top ranked suggestion as g_{ij};

Construct the new proposal: $\boldsymbol{g}_i \leftarrow \{g_{i1}, g_{i2} \cdots g_{im}\}$;

Choose w suggested CAMs to piggyback;

Broadcast its CAM by piggybacking $\boldsymbol{g}_i$ and the selected w CAMs;

```
Upon receiving CAMs;
```

for *each* $\mathbf{g}_i$ *in* $\mathbf{cam}_j$ **do**

  Cache each suggestion in $\boldsymbol{rs}_i$

for *each item in* $\mathbf{cam}_j$ **do**

  if *there is no such CAM in* B_i **then**

    store it in B_i.

  else

    keep the copy with the latest timestamps.

```
Upon CAM deadline expires:
```

Discard the expired CAM.

proposal, say $\boldsymbol{g}_c = \{g_{ca} = m_a, g_{ab} = m_b, g_{cc} = m_a\}$, and piggybacks m_a. Then the lost CAM m_a will be recovered.

Analysis on Convergence Speed: We define $\boldsymbol{F}^*$ as the optimal solution in MLP, and $\phi(t) = P\{u \leq t \mid \boldsymbol{F}(t) = \boldsymbol{F}^*\}$ is defined as the probability of all the vehicles achieve $\boldsymbol{F}^*$ for the first time before the tth piggybacking, and then we have $\lim_{t \to +\infty} \phi(t) = 1$ and Eq.(3) as below.

$$
\begin{aligned}
\mathbf{E}(u) &= \sum_{t=0}^{+\infty} t \cdot P\{u = t\} = \sum_{t=0}^{+\infty} t \cdot (\phi(t) - \phi(t-1)) \\
&= 1 \cdot (\phi(1) - \phi(0)) + 2 \cdot (\phi(2) - \phi(1)) \cdots \\
&= \sum_{i=1}^{+\infty} (\sum_{t=i}^{+\infty} (\phi(t) - \phi(t-1))) \\
&= \sum_{i=1}^{+\infty} \left[\lim_{t \to +\infty} \phi(t) - \phi(i) \right] = \sum_{i=1}^{+\infty} (1 - \phi(i))
\end{aligned} \tag{3}
$$

We can induce the expectation of the convergence speed $\mathbf{E}(u)$ based on $\phi(t)$ in Eq.(3). However, the problem is that it is not easy to calculate $\phi(t)$ in the piggybacking. Alternatively, it will be easier to evaluate $\mathbf{E}(u)$ if we can find an upper and lower bound of $\phi(t)$.

Let $P\{\boldsymbol{F}(t) = \boldsymbol{F}^* | \boldsymbol{F}(t\text{-}1) \neq \boldsymbol{F}^*\} = \lambda(t)$, and the lower and upper bound of $\lambda(t)$ at time t is $a(t)$ and $b(t)$ respectively. We can have $\phi(t) = \lambda(t) \cdot [1 - \phi(t-1)] + 1 \cdot \phi(t-1)$. Further more, we can get $1 - \phi(t) = (1 - \phi(t-1)) \cdot (1 - \lambda(t)) \leq (1 - a(t) \cdot (1 - \phi(t-1))) = \prod_{i=0}^{t} (1 - a(i))$. Similarly, we also have $1 - \phi(t) \geq \prod_{i=0}^{t} (1 - b(i))$. Hence, the expectation in Eq.(3) can be induced as $\sum_{t=0}^{+\infty} \prod_{i=0}^{t} (1 - b(i)) \leq \mathbf{E}(u) \leq \sum_{t=0}^{+\infty} \prod_{i=0}^{t} (1 - a(i))$. Suppose that both $a(i) = a$ and $b(i) = b$ where a and b are both constants, $\mathbf{E}(u)$ can be simplified as $b^{-1} \leq \mathbf{E}(u) \leq a^{-1}$.

V. Simulations

In this section, we evaluate our schemes through simulations using real V2V communication traces given in the gatech/vehicular dataset [2], and compare our schemes with the following two schemes:

Retransmission based Forwarding (RF): Any vehicle that lost CAMs will request the sender to retransmit the CAMs. This broadcast scheme is simple but commonly used as a benchmark to evaluate the performance of broadcast.

Location based Retransmission (LR) [9]: A forwarder is selected based on geographic information. The central point of all the vehicles that broadcast the request for lost CAMs is firstly calculated based on the geographic information, and then the vehicle nearest to the central point will be selected as the forwarder to retransmit the lost CAMs.

A. Simulation Setup

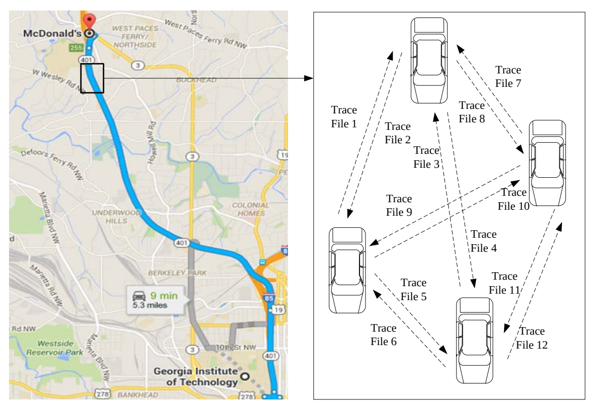

Fig. 3. Extend the V2V Communications in the Simulation

The V2V trace files in the gatech/vehicular dataset contains information including GPS location, speed, sending and receiving timestamps. Different trace files have different configuration on the V2V distance. In our simulations, we use multiple V2V traces to simulate CAM broadcast. Fig. 3 shows the trace setup for a VANET with 4 vehicles. Each channel link is allocated with a trace file. In the simulation, each vehicle can check the packet received state (i.e. lost or received) in the trace files according to the current geographic information.

To make fair comparisons, we compare the performance of different broadcast schemes with the same configuration: all simulation runs start at (33.793176N,84.390636W) and end at (33.824549N,84.424085W). Each v_i is allocated with a 2ms slot to broadcast CAMs, and the lifetime for each CAM is 100ms. For GP and MLP, each v_i can only piggyback one CAM in its scheduled time slot. Each vehicle generates a new CAM in every 100 milliseconds.

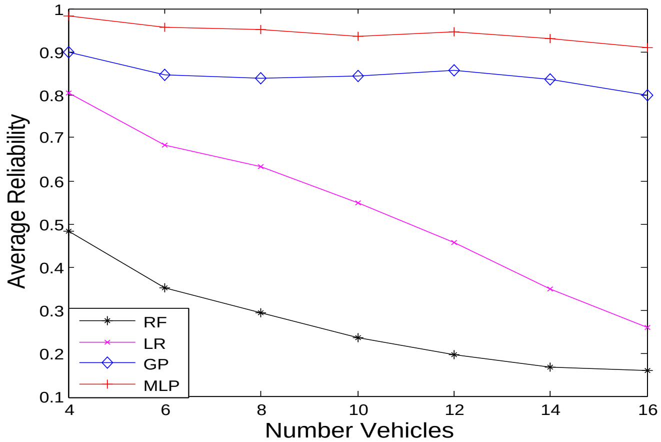

Fig. 4. Broadcast Reliability Against Different Vehicle Numbers

B. Comparison with Existing Schemes

Fig. 4 gives an overview of the average reliability by varying the number of vehicles from 4 to 16. It can be seen that GP and MLP can achieve much better broadcast reliability (most above 80%) than RF and LR. No surprisingly, the average reliabilities in LR and RF decrease significantly against the increase of the number of vehicle as more and more CAMs cannot be recovered with the increase of vehicles numbers.

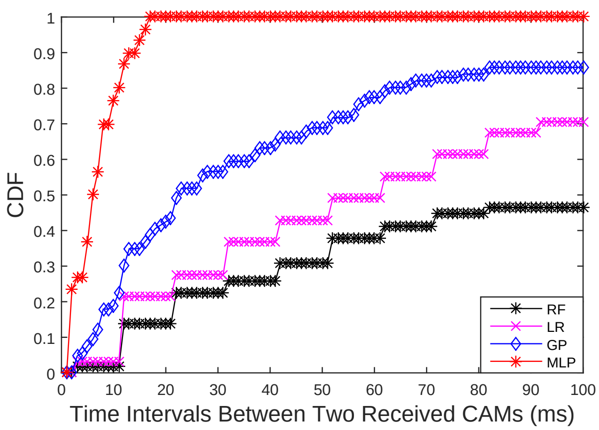

Fig. 5. Cumulated distribution of time intervals between two adjacent received CAMs.

Fig.5 compares the cumulated distribution of the time intervals between two adjacent received CAMs in the four broadcast schemes with 6 vehicles. Followed by GP, MLP has the best performance as all the time intervals between two received CAMs are always smaller than 15ms. The reason is because vehicles in MLP can obtain more information from the received proposals, which makes the piggybacking more efficient. The cumulated distribution in GP climbs to 0.88 at the end of CAMs's deadline(100ms), which is much higher than LR at 0.7 and RF at 0.48 respectively. According to DSRC, the deadline for each CAM is 100 milliseconds. The topology of the VANET hardly changes and the signal blocks or noise can still exist in such a short time interval. That explains why the simple retransmission in the RF scheme has the worst performance. Comparing with RF, geographic information is taken into consideration in the forwarder selection in LR. The optimal forwarding position is calculated and the vehicle nearest to this position is selected. Hence, signal blocks and noises can be bypassed in some degree.

VI. Conclusion

This paper proposes two decentralized broadcast schemes, GP and MLP, to solve the reliable CAM broadcast problem in the VANET. All the vehicles in the VANET piggyback some received CAMs cooperatively with expectation that the lost CAMs will be recovered after the piggybacking. Simulation results demonstrate that the proposed schemes can achieve much higher broadcast reliability compared with existing solutions. Our future work is to further improve the convergence speed, that is, minimize the iterations in the piggybacking, to reduce the overhead in the decentralized cooperative piggybacking.

Acknowledgements

The authors would like to thank the Natural Science Foundation for Colleges and Universities in Jiangsu Province (15KJB580009) and the Nature Science Foundation of Jiangsu Province (BK20130977), for partly funding this project.

References

[1] Araniti G, Campolo C, Condoluci M, et al. LTE for vehicular networking: a survey[J]. Communications Magazine, IEEE, 2013, 51(5): 148-157.

[2] Richard M. Fujimoto, Randall Guensler, Michael P. Hunter, Hao Wu, Mahesh Palekar, Jaesup Lee, Joonho Ko, CRAWDAD dataset gatech/vehicular (v. 2006?03?15), doi:10.15783/C74S3Z, Mar 2006.

[3] Lloret J, Canovas A, Catal A, et al. Group-based protocol and mobility model for VANETs to offer internet access[J]. Journal of Network and Computer Applications, 2013, 36(3): 1027-1038.

[4] Ma X, Zhang J, Wu T. Reliability analysis of one-hop safety-critical broadcast services in VANETs[J]. Vehicular Technology, IEEE Transactions on, 2011, 60(8): 3933-3946.

[5] Taha M M I, Hasan Y M Y. VANET-DSRC protocol for reliable broadcasting of life safety messages[C]//Signal Processing and Information Technology, 2007 IEEE International Symposium on. IEEE, 2007: 104-109.

[6] Kaul S, Yates R, Gruteser M. On piggybacking in vehicular networks[C]//Global Telecommunications Conference (GLOBECOM 2011), 2011 IEEE. IEEE, 2011: 1-5.

[7] Ma X, Zhang J, Yin X, et al. Design and analysis of a robust broadcast scheme for VANET safety-related services[J]. Vehicular Technology, IEEE Transactions on, 2012, 61(1): 46-61.

[8] Lou W, Wu J. Toward broadcast reliability in mobile ad hoc networks with double coverage[J]. Mobile Computing, IEEE Transactions on. 2007, 6(2): 148-163.

[9] Liu J, Yang Z, Stojmenovic I. Receiver consensus: on-time warning delivery for vehicular ad-hoc networks[J]. Emerging Topics in Computing, IEEE Transactions on, 2013, 1(1): 57-68.

[10] Sharma S, Teneketzis D. Local public good provisioning in networks: A Nash implementation mechanism[J]. Selected Areas in Communications, IEEE Journal on, 2012, 30(11): 2105-2116.