DUPE: Duplicated Unicast Packet Encapsulation in Position-Based Routing VANET

Ye Tao [*], Xin Li [†], Manabu Tsukada [*], Hiroshi Esaki [*]
[*] Graduate School of Information Science and Technology, The University of Tokyo
[†] School of Information and Communication Engineering, Beijing University of Posts and Telecommunications
[*] {tydus,tsukada}@hongo.wide.ad.jp, hiroshi@wide.ad.jp, [†] leexin@bupt.edu.cn

***Abstract*—Position-based Routing Vehicular Ad-hoc Network (PBR-VANET) is a specific type of mobile Ad-hoc network where the nodes are present as vehicles and the routing is decided based on node position. However, PBR-VANET routing protocols greatly suffer from mass packet loss due to frequent changes in topology. We discuss the requirements of the routing protocol and propose a routing encapsulation protocol called Duplicated Unicast Packet Encapsulation (DUPE). The protocol is compatible with original GeoNetworking standard, and can cooperate with the existing implementations without modification to neither GeoNetworking infrastructure nor user applications. The protocol encapsulates GeoNetworking packet in BTP packet (a Layer 4 protocol in GeoNetworking) and duplicates the original packet into different paths to reduce packet loss caused by failure and stale paths. The evaluations on DUPE in different scenarios show that, with reasonable latency and bandwidth overhead achieved, packet loss in certain scenarios is eliminated, where standard GeoNetworking protocol has packet loss of up to 94.5% in worst case.**

***Index Terms*—Vehicular Ad-hoc Networks, Cooperative Intelligent Transportation Systems, GeoNetworking, Routing Protocols, Open Source Software**

I. Introduction

Road transportation is one of the most essential services for human mobility since ancient times. Intelligent Transportation Systems (ITS) aim at optimization of the road traffic by realizing safe, efficient and comfortable transportation. Within a number of research fields in ITS, Application of Cooperative ITS includes driver assistance shortly. However, the vehicular communication also remains essential in autonomous driving to support a wider perception of the other vehicles around a vehicle that cannot be detected by the sensors equipped with the vehicle.

To communicate among vehicles and roadside units, GeoNetworking [1] is employed as one of the network protocols in the ITS Station architecture [2] standardized in ISO and ETSI, as shown in Figure 1.

GeoNetworking employees Position-based routing VANET (PBR-VANET) to adapt high-speed movement requirement. PBR-VANET is a type of VANET routing protocols which uses position information of nodes to direct routing. They do not maintain routing tables and exchange link state information but realize the route by the position of the nodes. This type of routing protocols performs better in high dynamic topology while link states are frequently changing.

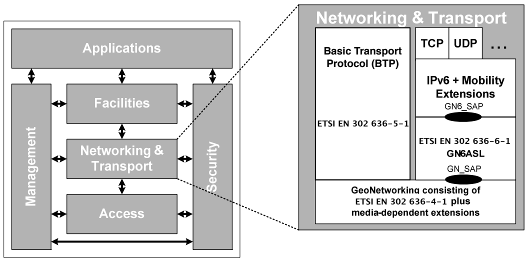


Fig. 1. GeoNetworking in ITS Station Architecture

Ensuring on-time delivery of packets is crucial in Intelligent Transportation Services. However currently VANET routing protocols are suffering from mass packet loss (up to 94.5% in certain cases). Many applications are sensitive to reliable communication, especially the safety-related ones. In extreme cases, humans' life may be threatened by the non-delivery of messages. Packet delivery ratio is the main challenge in VANET routing protocols. A frequently loss link could render many applications meaningless.

The rest of the paper is organized as follows. In section II, we analyze the packet loss issue we encountered in PBR-VANETs with two typical scenarios, *the overtake scenario* and *the shadow scenario*. As results, two reasons are derived from the scenarios. In section III we depict some efforts to solve the issue, which is the existing routing protocols for VANETs and an open source software of GeoNetworking. After that, additional requirements and constraints to each of the issues were discussed. In section IV, we discuss the additional requirements and constraints of the routing protocol, then propose a routing encapsulation layer called **Duplicated Unicast Packet Encapsulation (DUPE)** to solve the packet loss issue. In section V, several experiments of DUPE is conducted by the evaluation method of Combined Realistic Evaluation Workflow (CREW) , which is proposed in [3]. All of them show a fairly promising result: we efficiently eliminated the packet loss, with a reasonable overhead. Finally, in section VI, we summarize our contributions regards DUPE and make a prospect to future work.

II. Problem statement

In this paper, we mainly focus on two types of reasons which could cause packet loss on `PBR-VANETs`.

A. Adaptation to topology change

Most of the `PBR-VANET` routing protocols are working based on `beacons`, to maintain a constant update of position information about other nodes in the network.

In a realistic situation, the speed of vehicles can be changed substantially due to brake or acceleration. This feature leads to a highly dynamic topology, of which the change can happen in several hundred milliseconds. In the scenario depicted in Figure 2 (named **the overtake scenario**), when a vehicle *Bob* accelerates to overtake *Carol*, the original topology changes along with the overtaking. But *Alice* may be not aware of the topology change instantly, if there is not a dense communication between *Alice* and *Bob*. Messages from *Alice* to *Bob* are, therefore, all lost, as *Alice* holds stale position information about *Bob*, making *Alice* trying to reach *Bob* directly ignoring *Carol*, until *Alice* gets the updated information about *Bob*.

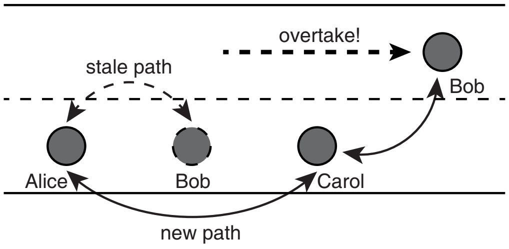


Fig. 2. The overtake problem

B. Recover from failure path

Under real environment, the wireless communication channel can be interfered by many factors. Such as buildings, radio noise. An example scenario is depicted in Figure 3, which is called **the shadow scenario**. If the communication channel between *Alice* and *Bob* is unstable, caused by the long transmission range here, but not lossy enough to lose all messages. Some sporadic messages from *Bob*, which carries the position information of it, can be transmitted successfully to *Alice*. Those messages make *Alice* choose to deliver the message directly to *Bob* without considering packet loss ratio and another alternative route. This scenario can exist in part of the route path of `PBR-VANET`, and result in high packet loss ratio. Most of the `PBR-VANET` routing protocols cannot perceive lossy stretches of the route, like the one between *Alice* and *Bob*, not to mention altering them with stable ones, with *Carol* relaying messages in this specific scenario. This leads to a *shadow area* for *Alice*, in which the packet loss ratio can be greater than 70%.

III. Related works

A. VANET routing protocols

Both Mobile Ad hoc Networking and Vehicular Ad hoc Networking work based on the same communication infrastructures – IEEE 802.11 Wireless LAN. Nevertheless, comparing to Mobile Ad hoc networks, VANETs have different resources to be employed, different targets to be achieved, and different problems to be solved.

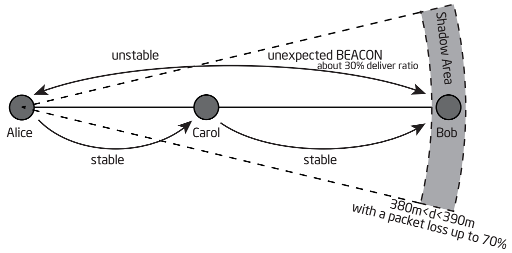


Fig. 3. The shadow problem

The special requirements of VANET are majorly classified into four categories: highly dynamic topology, scalability, utilization GPS and sensors and adaption to different environments. To satisfy the requirements, many efforts have been down to develop routing protocols for VANET. Some of the existing protocols for MANET are adapted to VANET while some dedicated ones were proposed to address the special challenges of VANET. [4] classified existing works of vehicle-to-vehicle routing protocols, which includes topology based, position based and other types.

Greedy Forwarding (GF) [5] is a simple protocol. In the protocol, each node maintains a neighbor table by collecting received beacons. Each neighbor table entry will time-out if no more beacon is received over a period of time. When a node needs to send a message to a node which is not directly reachable, it selects the neighbor who has the shortest Euclidean distance towards the destination. If the neighbor is not closer than the node itself, the algorithm fails and indicates it comes to a local maximum. GF will fail if a concave routing void exists between the node and destination.

Greedy Perimeter Stateless Routing (GPSR) [5] is an extended algorithm beyond GF, which could recover from local maximum caused by a routing void. GPSR defined two working modes: one is Greedy Mode as referred above. The other is named perimeter mode, which is triggered when greedy mode fails, which selects the neighbor using *Right-hand Rule* and help packet recover from local maximum.

Advanced Greedy Forwarding (AGF) [6] is an improvement to GF. In the traditional GF protocol, the position of the destination node in the packet is never updated. However in a high-speed movement scenario, during the packet traveling, the destination node itself also travels a significant distance. AGF could update the destination position information according to predictions by velocity given in packet. The experiment results show a triple improvement to packet delivery ratio in certain scenarios.

Greedy Traffic-Aware Routing (GyTAR) [7] is a junction-based routine protocol, which means efficiently choosing junctions which have a greater chance to meet and forward to different nodes. Junctions are scored by equation $S_j =$

$af(T_j) + bg(D_j)$ where T_j is the traffic density and D_j is the distance to destination. The junction of the maximum score is selected, and the packet is forwarded to it.

Contention Based Forwarding (CBF) [8], [9] is a protocol which does not employ neighbor information for geographical routing. Instead, the routing decision is made by the receiver but not the sender. The receivers cooperate by a predefined distributed timer: the first timeout node rebroadcasts the message which will suppress the others. CBF saves beacon consumption and also improves forwarding efficiency compared with flooding based approaches. However, it suffers from end-to-end latency since a packet could only be forwarded when a timeout occurs on either receiver.

B. Multipath routing

Multipath routing is an idea to duplicate the packet and take multiple paths for delivering. Multipath routing protocols are discussed in Mobile ad-hoc Networks, as well as in other fields like Wireless Sensor Networks, VoIP, and IPTV. According to the motivation of the protocols, multipath routing protocol could be classified into three types: reliable communication, delay minimization, and load balancing.

Ad-hoc On-demand Multipath Distance Vector (AOMDV) [10] is an extension of AODV [11] with improved performances. When required, AOMDV selects multiple paths and ensuring they are disjoint and loop-free. AOMDV can make use of existing information from underlying AODV protocol to reduce probing overhead. AODV based multipath routing protocols could not get rid of the drawback of AODV in handling high-speed movements. Therefore, AODV could not best fit the requirements of VANET.

Static-node-assisted Adaptive data Dissemination for VANET (SADV) [12] is a multipath VANET routing protocol which utilizes pre-deployed static infrastructures. In SADV, a packet is forwarded to static nodes when no vehicles on the road could carry and forward the packet. The packet forwarded to a static node waits until a vehicle comes and forward it. SADV uses several different paths to forward the same packet to ensure it is delivered in low latency. SADV could provide multipath low latency communication. However, it relies on pre-deployed infrastructures, which may not be practical on all the roads.

C. GeoNetworking

GeoNetworking [1] is standardized by ETSI as a network layer protocol as in Figure 1, integrating several position-based routing strategies including Greedy Forwarding (GF) [5] (Also known as *GPSR*), which chooses a directly reachable node which is closest to the destination based on GPS location obtained by *Location Service(LS)* request action, to route packets better in vehicular networks. Above the GeoNetworking, there are two different layers. One is *Basic Transport Protocol (BTP)* [13] which provides basic functions of the transport layer to GeoNetworking, the other is *GeoNetworking to IPv6 Adaptation Sub-Layer (GN6ASL)* [14] to enable standard IPv6 over GeoNetworking. In GeoNetworking, three forwarding modes are defined based on the destination in the geographic scope (*GeoUnicast*, *GeoBroadcast*, and *GeoAnycast*), otherwise, there are two forwarding modes that does not use location information: *Topological Scope Broadcast (TSB)* to broadcast the packet within a specific hop away and *Single Hop Broadcast (SHB)* to broadcast the packet within the 1 hop neighbors. Packets destine the geographic scopes are delivered by either GF [5] or CBF [8].

All the GeoNetworking nodes send beacons in a specific interval, and the neighbor nodes maintain its latest geographical location in the *location table (LocT)* from the received beacons. Other GeoNetworking packets delivered in the network contain the location of *source (SO)*, *sender (SE)* and *destination (DE)*; in the case that the location information in the packet is newer than the one in the location table, the location table is updated. Each *location table entry (LocTE)* has a lifetime counter, and the entry is removed when it is reduced to 0. When the source node does not have location of the designation in its location table, the node triggers the *Location Service (LS)* request message in order to obtain the location of the destination. ETSI defines the flooding based request-reply location service to get the destination location.

D. CarGeo6: user space GeoNetworking protocol stack implementation

More than fifteen software implementations of GeoNetworking join ETSI Plugtest that provides interoperability test opportunity every year. In our best knowledge, only the CarGeo6 project provides GeoNetworking implementation in open source [15].

The GeoNetworking function and the BTP function are implemented as daemons called `itsnet` and `btpecho`, respectively in the CarGeo6 implementation as in the Figure 4. In source node, `btpecho` (client mode) sends a BTP packet via inter-process communication to `itsnet`. If the destination location is in `LocT`, `itsnet` forwards the packet to next hop selected by GF. Otherwise, it triggers an LS request. Finally, when the BTP echo request is forwarded to the destination, `itsnet` send the packet to `btpecho`. On the other hand, `btpecho` (reflector mode) in the destination node sends a BTP echo reply back to the source once it received a request. The echo reply is also forwarded by GF. Thus, the reply packet may be delivered via a different route from the request packet.

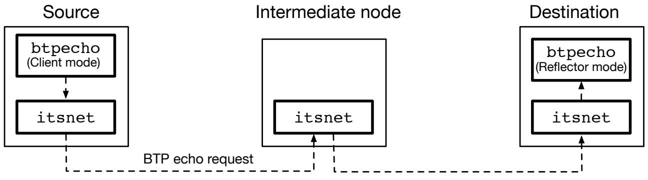


Fig. 4. Overview of CarGeo6 programs

IV. Duplicated Unicast Packet Encapsulation (DUPE) for GeoNetworking

To solve the packet loss issue caused by topology change and failure path, here we propose the Duplicated Unicast Packet Encapsulation (DUPE), a standard compatible drop-in encapsulation layer to the GeoNetworking standard architecture.

A. Requirements

Besides the problem, some additional requirements to the routine protocols besides Packet Delivery Ratio should also be considered.

1) **No changes on GeoNetworking Standards:** GeoNetworking is an essential part of ETSI standard ITS station architecture, which presents in the Networking and Transport Layer. GeoNetworking includes Network Layer and Transport Layer protocols and auxiliary facilities as depicted in Figure 1, while interfaces for interactive with different parts were also defined. The protocol should work according to the standard and do not require any extensions to GeoNetworking and other parts of ITS station architecture.

2) **No changes on user applications:** In GeoNetworking standard, several interfaces between GeoNetworking infrastructures and user applications were documented. In both parts, numbers of existing software and hardware are available, built by different developers. Maintain the compatibility with both part and require no changes to user application are essential for the protocol of being widely used.

3) **Deliver packets on-time:** Cooperative ITS need real time and on-time communication with other vehicles, to realize safety by being aware of emergencies as soon as possible. Thus in VANET, many applications are more strict in packet delivery ratio and maximum delay. For example information about an accident happened on a highway should be immediately delivered to the consequential automated vehicles on the road, since not all of them could be aware of the accident from their view. Certainly, accidents could not be expected, and vehicles could not prepare for them. Delivering the packet on time may save passengers' life.

4) **Moderate resource consumption:** The VANETs are essentially wireless networks and obviously have a limitation in bandwidth. In most cases, most of the nodes should have transit traffic which forwarded by other vehicles. Moreover, many communications may happen simultaneously causing more transit traffic; this could burden bandwidth of a single node furthermore. Nowadays, computing devices become more powerful. VANET devices are not an exception. However, the resources still limited towards the scalability in specific scenarios: CPU calculation power, memory limitation, etc. Both bandwidth and resource limitation should be better considered in the design of the protocol.

5) **Utilize alternative paths:** According to issue analysis, we know the issues causing packet loss in VANETs are stale paths and unstable paths. Nevertheless, at the same time, several different routing paths may exist, which are less optimal to the problematic one but more stable. We call such sub-optimal ones Alternative Paths (APs). The protocol should select and utilize APs efficiently is also a consideration to reliable delivering.

B. Approaches

Three approaches were figured out to fix the issues and fit requirements for routing protocol. As the name of the protocol, the approaches are: **Duplicated**, **Unicast**, **Packet Encapsulation**.

1) **Packet Encapsulation:** In the proposal, GeoNetworking packets (in Network Layer) are encapsulated into plain *BTP-B* packets. *BTP-B*[14] is a standard transport layer protocol defined in GeoNetworking standards, thus the *DUPE* packet could be processed normally in nodes, which satisfies the requirement defined in Section IV-A1 (no changes on GeoNetworking Standards). The packets in capsules (a.k.a. payload) are also GeoNetworking packets and could be processed either by the original GeoNetworking protocol handler (depends on implementation) or a separated daemon providing the same interface (i.e. *GN_SAP*[16]). The implementation should provide all the functionalities of *GN_SAP* interface, in order to maintain the compatibility with 3rd-party user programs. This satisfies the requirement in Section IV-A2 (no changes on user applications).

2) **Duplicate the packet:** The main issue to be solved by the protocol is packet loss. According to analysis, we know that packet loss of VANETs could be caused by two reasons: stale path and unstable path. Nevertheless, when the selected greedy path is stale or unstable, several alternative paths may also exist and could be more viable comparing to the selected ones. Thus, we decided to duplicate the packets in such cases, to utilize alternative paths (the requirement in Section IV-A5). The duplication of packets could also ensure that the requirement defined in Section IV-A3(on-time packet delivery) while timeout based approaches e.g. CBF or retransmission could increase packet latency.

3) **Limited duplication using one hop GeoUnicast:** One hop *GeoUnicast*[1] was employed to limit the number of duplication of a packet. The routing decisions are handled by *DUPE*, to bypass the routing selection in standard GeoNetworking layer, which could result in failing routing path. For each packet, *DUPE* selects several (depends on the significance of message, usually 2 or 3) next hop candidates and forward the DUPE packet by encapsulating it into one hop GeoUnicast, which means packet should directly send to next hop. Numbers of next hops were limited by *DUPE* to reduce bandwidth overhead, to meet the requirement of in Section IV-A4 (moderate resource consumption).

C. System overview

In GeoNetworking protocol stack, DUPE works between GeoNetworking layer and Transport layer, as depicted in Figure 5. DUPE encapsulation layer (with inner GeoNetworking layer) is an optional layer in the protocol stack, so user programs could either employ DUPE to achieve better packet

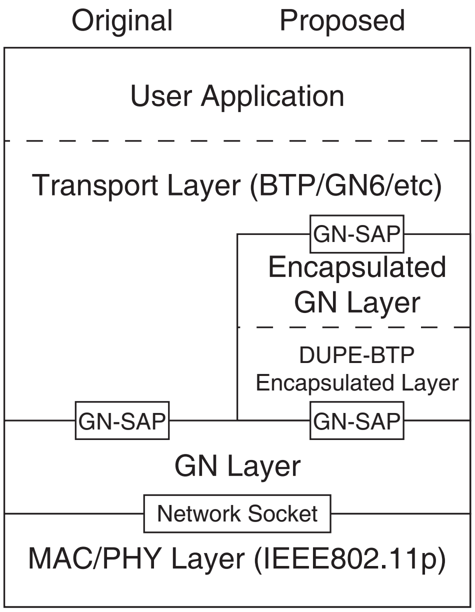


Fig. 5. GeoNetworking protocol stack and DUPE encapsulation layer

delivery ratio or come to ordinary GeoNetworking packet to save bandwidth.

The typical packet flow of *DUPE* is depicted in Figure 6. Firstly, a *Basic Transportation Protocol (BTP)* packet (which scope is *GeoUnicast (GUC)*) is generated by user application at *Source node*. The packet is required to be *DUPE* encapsulated, so it is prepended a *DUPE* header and becomes *DUPE* packet. *DUPE* processor selects two different intermediate nodes as candidates of next hops among its neighbors, and send the *DUPE* packet as payload of a *BTP* packet to them one by one.

The outer *BTP* packet is processed by standard GeoNetworking as a regular one and eventually reaches an intermediate node. The *DUPE* processor on *Intermediate node 1* decapsulates the packet, checks destination of the encapsulated packet and found it should be forwarded. Therefore, the processor encapsulates it into a new *BTP* packet, sets the destination of the outer packet to the real *Destination node*, and emits the packet. The *DUPE* processor on *Intermediate node 2* does the same thing, thus both packets from them come to *Destination node*.

The *DUPE* processor on *Destination node* decapsulates the first packet, found the destination matches local address, and send to the user application which should handle it. The second packet from the other *Intermediate node* is dropped by the *DUPE* processor to make sure user application receives only one copy of the packet.

Figure 7 shows how *DUPE* works in a network with obstacles, where standard GeoNetworking comes to a failure. When *A* tries to send a message to *F*, with standard GeoNetworking, as show in Figure 7(a), he will firstly send the message to *C* to let it relay. But the obstacle between *C* and *F* makes the message lost. When *DUPE* is adopted, as in Figure 7(b), *A* will duplicate the message to *B* and *C*. Although the message from *C* to *F* is lost, but *B* will pass on the message forward in the manner like the Figure 7(b) shows. Each node duplicates the message to best N next hops, here in Figure 7(b) the N equals 2.

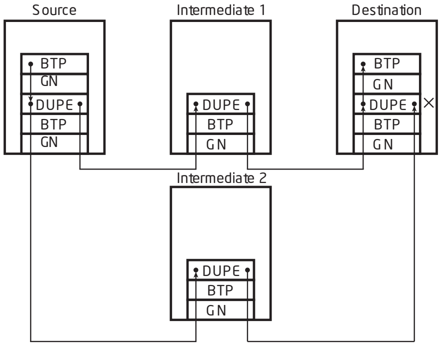
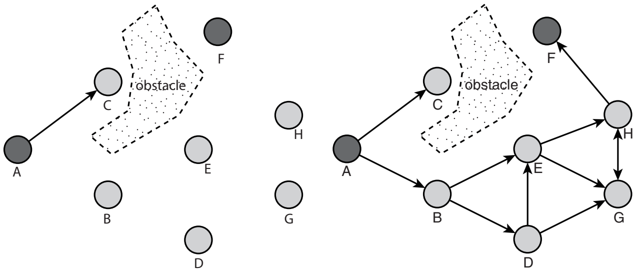


Fig. 6. DUPE packet flow



(a) GeoNetworking

(b) DUPE

Fig. 7. How DUPE dealing with obstacle

D. Implementation details

Because *DUPE* is designed as a drop-in layer, we implemented it as a separated daemon `duped`. Duped communicates with both sides (`itsnetd` and user programs) using libgnsap provided by CarGeo6. It handles information through the standard SAPs defined by GeoNetworking and provides the same interfaces to the upper layer. In the duped, tasks are divided into different modules, which is shown in Figure 8. Output module receives packet request from the user programs, encapsulates it into BTP packet and deliver it. Input module receives BTP packet from the lower layer, decapsulates it and notify the user program from GN_SAP interface. Deliver module duplicates the packet, select best next hops from candidates for each ones, and then send them to lower layer. Location module handles the location information used by different modules.

V. Evaluation

A. Experiment setup

In this paper, we conduct the experiments within the *Combined Realistic Evaluation Workflow (CREW)* [3]. The workflow contains three steps: experimentation, reproduction and extension. First, the experimental evaluations are conducted in

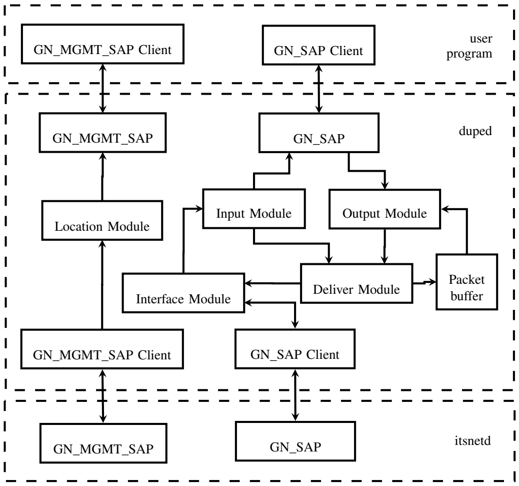


Fig. 8. Module diagram of duped and interactive with itsnet

the real testbed, and the results are collected. Next, the same evaluations with the same hardware environments and software implementations are reproduced in the simulated networks. We tune the hardware and software parameters of the simulation environment in order to calibrate the simulator using the results produced in the last step. Last, we extend the experiments in different scenarios and larger networks in the calibrated simulated environment, to produce realistic simulated results in a low cost.

In the experiment, we make a horizontal comparison between *DUPE* based software stack and *original CarGeo6* ones in the same environment and scenarios. On the side of the real testbed, the programs ran on 4 ARM boxes. Mac filters were applied to each node, to limit the neighbor that it could communicate with adjacent nodes only, achieving an up to 3 hop topology as depicted in Figure 9. In simulation, we use a linear topology which is shown in the Figure 9: all nodes are configured with the same Wi-Fi parameters, and kept in the same Ad-Hoc cell; each node is in a line with 300m distance to adjacent nodes. The wireless radio range is adjusted to 300-400 meters. That means nearly all packets in 300m range can be delivered, yet almost all packets from 400m away were lost.

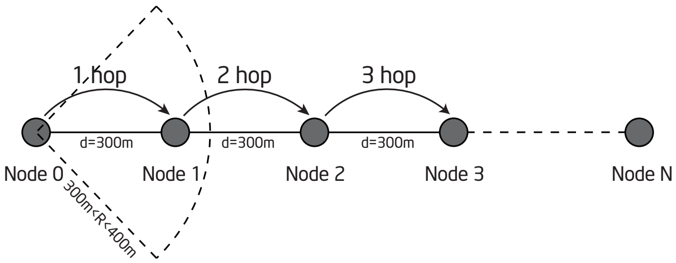


Fig. 9. Topology of 300m distance (achieved by mac filter)

The detailed system configuration of the simulator is depicted in Table I.

TABLE I
SIMULATOR SYSTEM CONFIGURATION

Item	Specification
Simulator	NS3-DCE
Radio Frequency	5900 MHz
Wi-Fi Phy	ERP-OFDM, 6 Mbps
Wi-Fi Mac	Ad-Hoc
Receiver Gain	-10 dBi
Propagation Delay	Constant
Propagation Loss	Friis
Node Mobility Model	Static

B. Evaluation scenarios

In the experiment, we vary node distances, packet sizes, and number of hops. We define Neighbor distance (*ND*) as the distance between adjacent nodes, Terminal Distance (*TD*) as the distance between destination and source nodes.

The first experiment we conducted is different *ND* and *TD*s. We measure 1000 round trip packets of `btpecho` between the first and last nodes, with fixed *ND* = 300 m and *TD* varied from 300 to 3000 meters.

The second experiment is about *the shadow problem* which is described in Section II-B. The *ND* is fixed to 100 meters, with varied *TD*s to make *the shadow problem* happen between different hops. The Network topology of the experiment is depicted in Figure 10.

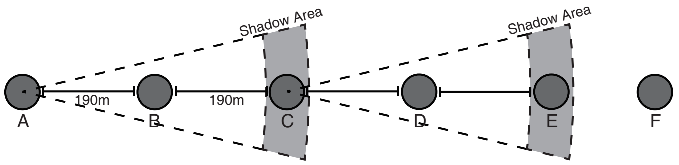


Fig. 10. Chain of shadow areas in 190 meter tests

C. Evaluation results

In the experiments, we evaluate the packet delivery from end to end. These three metrics are considered, prioritized from top to bottom.

1) **Packet delivery ratio (PDR):** PDR is the basic metric to measure the quality of a network link. Packet delivery is the most important requirement in VANETs even rather than latency. Thus, it is the most direct evaluation metric to measure the quality of `DUPE`.

Figure 11 shows the comparison of PDR of *DUPE* and *CarGeo6* under different *ND* and *TD*s. It suggests that *DUPE* can persist high PDR while *CarGeo6* is unstable in long distance.

Especially, when the *TD* is 190 meters, which leads to a scenario conducting *the shadow problem*, as we have discussed in Section II-B. Remind the topology (Figure 10) we use in this scenario. When *A* sends a message to *F*, it will firstly let *C* to relay the message, then *C* chooses *E* to relay the message to *F*. The *Shadow Area* seriously affects the PDR of messages from *A* to *C*, and messages from *C* to *E*. The

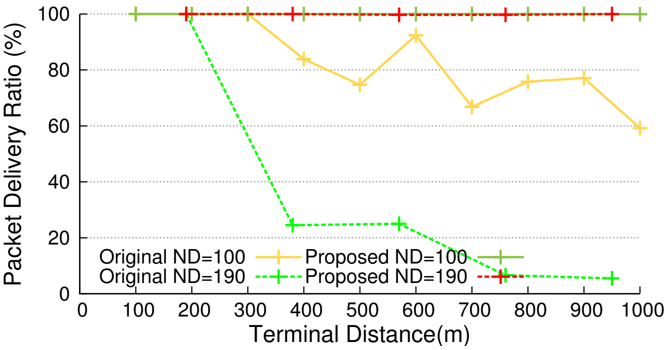


Fig. 11. Comparison of Packet Delivery Ratio in 100 meters and the shadow scenario

superimposition of packet losses results in about 5.5% of PDR, when the terminal distance is greater than 780 meters in this scenario. Meanwhile, the PDR of *DUPE* remains constantly high in contrast.

2) **Round-trip time (RTT):** RTT is also an important metric in network communication. In VANET, the maximum latency is more important because of the *on-time delivery* constraint from the applications.

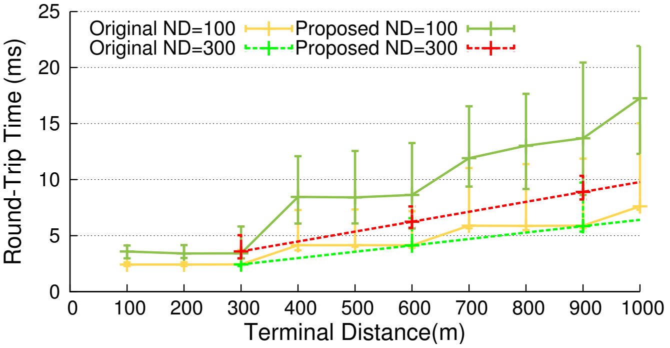


Fig. 12. Minimum-average-maximum round-trip Time

Figure 12 shows the minimum-average-max round trip time of *DUPE* in different *ND* and *TD*.

When $ND = 300$ m, both average and maximum *RTTs* show a tendency of linear growth on hop count. However, both of them are slower than the original *CarGeo6* because of additional packet processing. When $ND = 100$ m, both average and maximum *RTTs* are notably higher than same TD of $ND = 300$ m. This is possibly due to additional packet processing and interference caused by multiple time transmissions. A deeper profiling and analysis should be done to locate the specific reasons leads to extra latency. Although *DUPE* introduced some overhead in packet processing, the maximum *RTT* is reasonable: around 22 miliseconds in 1000 m experiment.

3) **Transmission cost (TC):** TC denotes totally how many packets are transmitted to deliver one packet of data successfully from end to end, includes lost ones and unused ones. Reduce *transmission cost* could save bandwidth consumption and get rid of packet loss caused by congestion or wireless interference. Transmission cost of program p (TC_p) is given by Equation 1, where T_i refers to GeoNetworking packet transmitted on node i, NT_p refers to total transmitted packets of program p, and ND_p denotes number of packet successfully delivered.

$$TC_p = \frac{\sum_{i \in Nodes_p} T_i}{Nd_p} = \frac{NT_p}{ND_p} \tag{1}$$

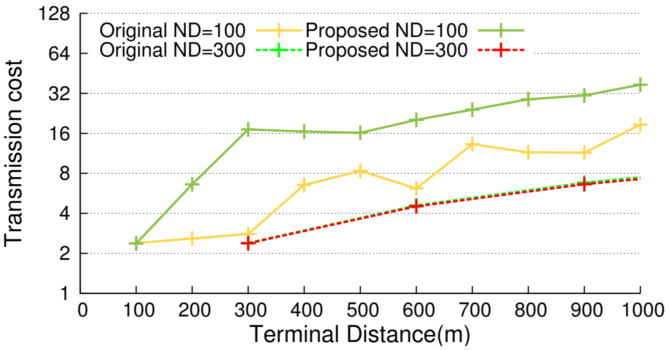


Fig. 13. Transmission cost of DUPE and CarGeo6

Figure 13 shows *TC* of *DUPE* compare with *CarGeo6*, in 100 meter experiments, with different *TD*.

In the experiments, we evaluate TC_{DUPE} divided by $TC_{CarGeo6}$, which means how many times of overhead *DUPE* is committed over the original *CarGeo6*.

In $ND = 300$ m experiments, the *TC* of *DUPE* and *CarGeo6* are almost the same, suggests the transmission of *DUPE* in this scenario is minimal. The reason is the nodes are sparse, and each one could only connect to two neighbors: receive from one of them, and send to another.

While in $ND = 100$ m, the results varies by TD. When $TD = 100$ m, which means only two vehicles in the test, *TC* is also the same. The reason is as same as $ND = 300$ m.

When $TD = 300$ m, *DUPE* is high because of backward propagations, which means to forward packets to the node which is farther than the node itself. If backward propagation is disabled, the overhead could be reduced to around 2 times.

When $TD >= 400$ m, the *PDR* reduces, which means many transmissions is wasted because of packet loss. Thus, the overhead reduced again to less than 2, suggests it could deliver packet more efficiently.

VI. Conclusion and future work

ITS is a technique to realize safe, efficient and comfortable driving experiences. VANET is a technique for direct inter-vehicular communication, which is essential for *ITS*. However current a widely deployed VANET standard, which is *GeoNetworking*, faces the serious packet loss problem in dynamic topology. The packet loss problem must be solved because it could render safety related applications completely useless.

In this paper, we proposed the Duplicated Unicast Packet Encapsulation (DUPE) protocol to enable reliable real-time communication among vehicles for VANETs.

DUPE was evaluated on *CREW*, suggesting that packet loss in different scenarios can be successfully eliminated, especially in *the shadow problem* and *the overtake problem*.

The main features of *DUPE* are:

1) Reliable
   Experiments about *DUPE* shows it eliminated packet loss in both *the shadow scenario* and *the Overtake scenario*. Among them, *CREW* evaluation indicates in the worse case of *the shadow scenario*, could increase end to end *Packet delivery Ratio* from **5.5%** to nearly **100%**. These two simple and typical scenarios are likely to happen in the real VANETs. Generally, *DUPE* could increase the *packet delivery ratio (PDR)*, leads to a more reliable communication channel.
2) Independent
   DUPE is designed to maintain independence in different sense: *DUPE* does not require modifications on GeoNetworking infrastructures; *DUPE* does not require modifications on user applications; *DUPE* does not require *digital maps* present on Nodes.
3) Efficient
   DUPE has a reasonable processing and duplicated transmission overhead. In short distance and dense traffic situations, *DUPE* performed well which means overhead is moderate. In long distance or sparse traffic scenarios, *DUPE* has a large duplication overhead; however *DUPE* could drastically increase *PDR* which makes most sense.

Due to time limitation of the master course, not all works were included in the paper. In the future, several directions are considered, while some of them are working in progress.

From the experiments and analysis, we know packet transmission cost of *DUPE* could be high in certain scenarios. To reduce *Transmission Cost*, mechanisms such as single hop broadcasting or adaptive duplication count could be considered. Such techniques may adjust according to communication distance and traffic density, to balance PDR with transmission overhead.

Currently, middleboxes (i.e. the nodes which did not run `duped`) could not be detected. For this issue, we need to introduce some mechanism to detect the middlebox. A reactive method could make sense because the status of "running" or "not-running" duped on a node may not change frequently. Another method of combining it with Multiple Hop Broadcast (MHB) is also considerable.

In this work, we start the DUPE based on GeoNetworking and GF algorithm, yet actually, it could be considered in any other beacon based position routing protocols. In the future, we could adapt DUPE to other protocols. Moreover, if the underlying protocol could provide more information e.g. speed and heading information of vehicles or digital maps, more advanced routing selection could be employed in DUPE.

References

[1] *Intelligent Transport Systems (ITS); Vehicular Communications; GeoNetworking; Part 4: Geographical addressing and forwarding for point-to-point and point-to-multipoint communications; Sub-part 1: Media-Independent Functionality*, Std., Jul. 2014. [Online]. Available: http://www.etsi.org/deliver/etsi_en/302600_302699/3026360401/01.02.01_60/en_3026360401v010201p.pdf

[2] *ISO 21217:2010 Intelligent transport systems – Communications access for land mobiles (CALM) – Architecture*, ISO CALM TC204 Std., April 2010.

[3] Y. Tao, M. Tsukada, X. Li, M. Kakiuchi, and H. Esaki, "Reproducing and extending real testbed evaluation of geonetworking implementation in simulated networks," in *The 10th International Conference on Future Internet*. ACM, 2015, pp. 27–34.

[4] B. T. Sharef, R. A. Alsaqour, and M. Ismail, "Vehicular communication ad hoc routing protocols: A survey," *Journal of network and computer applications*, vol. 40, pp. 363–396, 2014.

[5] B. Karp and H. T. Kung, "Gpsr: Greedy perimeter stateless routing for wireless networks," in *6th Annual International Conference on Mobile Computing and Networking, MobiCom 2000, August 6.-11., 2000, Boston, Massachusetts, USA*. ACM / IEEE, August 2000, pp. 243–254.

[6] V. Naumov, R. Baumann, and T. Gross, "An evaluation of inter-vehicle ad hoc networks based on realistic vehicular traces," in *Proceedings of the 7th ACM international symposium on Mobile ad hoc networking and computing*. ACM, 2006, pp. 108–119.

[7] M. Jerbi, S.-M. Senouci, R. Meraihi, and Y. Ghamri-Doudane, "An improved vehicular ad hoc routing protocol for city environments," in *Communications, 2007. ICC'07. IEEE International Conference on*. IEEE, 2007, pp. 3972–3979.

[8] H. Fußler, J. Widmer, M. Käsemann, and M. Mauve, "Contention-based forwarding for mobile ad hoc networks," *Ad Hoc Networks*, 2003.

[9] H. Füßler, H. Hartenstein, M. Mauve, W. Effelsberg, and J. Widmer, "Contention-based forwarding for street scenarios," in *1st International workshop in intelligent transportation (WIT 2004)*, no. LCA-CONF-2004-005, 2004.

[10] M. K. Marina and S. R. Das, "Ad hoc on-demand multipath distance vector routing," *Wireless Communications and Mobile Computing*, vol. 6, no. 7, pp. 969–988, 2006.

[11] C. Perkins, E. Belding-Royer, and S. Das, "Ad hoc on-demand distance vector (aodv) routing," Tech. Rep., 2003.

[12] Y. Ding, C. Wang, and L. Xiao, "A static-node assisted adaptive routing protocol in vehicular networks," in *Proceedings of the fourth ACM international workshop on Vehicular ad hoc networks*. ACM, 2007, pp. 59–68.

[13] *Intelligent Transport Systems (ITS); Vehicular Communications; GeoNetworking; Part 5: Transport Protocols; Sub-part 1: Basic Transport Protocol*, Std., Aug. 2014. [Online]. Available: http://www.etsi.org/deliver/etsi_en/302600_302699/3026360501/01.02.01_30/en_3026360501v010201v.pdf

[14] *Intelligent Transport Systems (ITS); Vehicular Communications; GeoNetworking; Part 6: Internet Integration; Sub-part 1: Transmission of IPv6 Packets over GeoNetworking Protocols*, Std., May 2014. [Online]. Available: http://www.etsi.org/deliver/etsi_en/302600_302699/3026360601/01.02.01_60/en_3026360601v010201p.pdf

[15] T. Toukabri, M. Tsukada, T. Ernst, and L. Bettaieb, "Experimental evaluation of an open source implementation of IPv6 GeoNetworking in VANETs," in *ITST 2011 : 11th International Conference on Intelligent Transport System Telecommunications*, Saint-Petersburg, Russia, Aug. 2011.

[16] *Intelligent Transport Systems (ITS); Vehicular Communications; GeoNetworking; Part 3: Network Architecture* , Std., Dec. 2014. [Online]. Available: http://www.etsi.org/deliver/etsi_en/302600_302699/30263603/01.02.01_60/en_30263603v010201p.pdf