A Distributed Taxi Hailing Protocol in Vehicular Ad-Hoc Networks

Jang-Ping Sheu*, Guey-Yun Chang†, and Chiung-Hung Chen†
*Department of Computer Science, National Tsing Hua University, Taiwan
†Department of Computer Science and Information Engineering, National Central University, Taiwan

***Abstract*—In this paper, a distributed taxi hailing protocol in vehicular ad-hoc networks (VANET) is suggested. Our protocol consists of two parts: taxi booking and taxi de-blocking. Taxi booking part ensures that a vacant taxi with shortest driving distance to passenger under real traffic regulations is booked. Taxi de-blocking part aims to de-block blocked vacant taxis as soon as possible. Simulation results show that compared with previous results, at least 40% of booking time, 50% of waiting time of passengers and 50% of driving distance from booked taxi to passenger, is reduced in our protocol.**

I. INTRODUCTION

In urban city, taxis provide convenience in middle distance travel. However, taxi markets face difficult challenge from high fuel cost which reduces profit margin for taxi companies [4, 6]. In order to reduce the fuel cost, efficient taxi hailing systems are necessary.

In this paper, we propose a distributed taxi hailing protocol in VANET. Our protocol consists of two parts: taxi booking and taxi de-blocking. In order to reduce fuel cost, the former part ensures that a vacant taxi with shortest driving distance to passenger under real traffic regulations is booked. The latter part aims to reduce the blocking time of taxis, i.e., de-blocking blocked vacant taxis as soon as possible. Compared with the distributed protocol EZCab [1] through simulations, the performance of our protocol is better than EZCab in terms of booking time, waiting time of passengers, driving distance from booked taxis to passenger and blocking time of taxis.

The rest of this paper is organized as follows. In Section II, we review some related works. In Section III, we present our distributed taxi hailing protocol in VANET. The simulation results are presented in Section IV. Finally, we conclude the paper in Section V.

II. THE PROPOSED TAXI HAILING PROTOCOL

In this section, we propose a distributed taxi-hailing protocol for urban areas. Throughout this paper, vehicles are assumed to be equipped with car computers, wireless transmission modules like 802.11, and GPS receivers, allowing car computers to calculate their coordinates or determine their location by matching the electronic maps. We also assume that it would be on main streets where taxi drivers pick up passengers or passengers wait for taxis. Since there is sufficient traffic volume on main streets, the connectivity of VANET is guaranteed [3, 5, 7].

In [1], taxis which are geographically closest to passengers are chosen and booked to serve passengers. However, booked taxis may have long driving distance to passengers. In Fig. 1, taxi *B* is the geographically closest to passenger *P*. However, in general, taxi *A* approaches passenger *P* more quickly than taxi *B* because the driving distance of taxi *A* is shorter than taxi *B*'s (taxi *A* picks the passenger up without making U turns like taxi *B*). Booking taxi *B* not only wastes the passenger's time but also increases the fuel consumption.

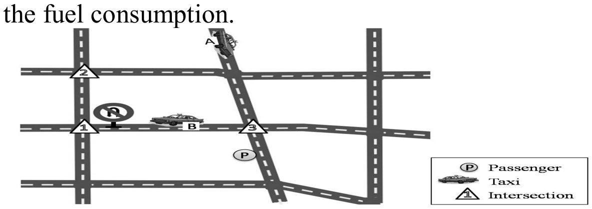

Figure 1. Taxi *B* is closer to passenger *P* than taxi *A*. But taxi *B* has longer driving distance to *P* than taxi *A*.

In this paper, we propose a distributed taxi hailing protocol in VANET. Our protocol consists of two parts: taxi booking and taxi de-blocking. In order to reduce fuel cost, taxi booking part ensures that a vacant taxi with shortest driving distance to passenger under real traffic regulations is booked. In order to guarantee that no two passengers book a taxi simultaneously, taxis competing for one passenger are blocked from competing for others when competition results are not available for them. Taxi de-blocking part de-blocks blocked vacant taxis as soon as possible. Taxi de-blocking provides more chances to vacant taxis for picking up passengers.

A. Taxi Booking

Taxi booking is a three-way handshake protocol. Passengers who need to hail taxis use their handheld devices with wireless transmission modules to transmit a request message (REQ) via VANETs. If a vacant taxi receives this message, the driver will send a reply message (REP) back to the passenger to compete for this job. After receiving the first reply, the passenger responds an ACK message to the taxi sending the first reply.

Taxis are sorted into vacancy, contention, and occupied modes. Vacant taxis enter the contention mode when receiving REQ messages from passengers. Taxis in contention mode determine their driving paths (on the main streets) from their current position to the passenger and calculates their driving distances. Notice that REP messages are transmitted along the determined driving paths from taxis to the passenger. Since the connectivity of VANET is guaranteed, taxis whose REP message first received by the passenger have the shortest driving distance to the passenger. When a passenger receives the first REP message from a taxi, the passenger sends the taxi a confirmation ACK.

In order to guarantee no two passengers book the same taxi simultaneously, taxis competing for one passengers are blocked to competing for others. Taxis are de-blocked when the competition result is available for them. Clearly, long blocking time results in taxis losing chances to pick up other passengers. Hence, blocking time should be reduced, i.e., taxis should be released as soon as possible. The detail of de-blocking blocked taxis can be found in Section III.B.

In our scheme, we aims to reduce not only fuel cost of taxis (by allowing taxis with the shortest driving distance to win the competition of picking up passengers), but also the waiting time for passengers to get taxis (by booking taxis which sent replies first received by passengers). Our protocol is detailed below.

Message packet format used in our protocol is assumed to have five fields as shown in Fig. 2. The Type field records the message types: REQ, REP, or STA. The Sender position field records the location of transmitter/forwarder and the Destination position field is the location which the packet is supposed to be sent to. The Sender position field and the Destination position field are used to determine the next forwarder of the packet. The Payload field records the main content of the message.

Type	Sender position	Destination position	Payload
1 byte	8 bytes	8 bytes	14~35 bytes

Figure 2. Packet format.

In Fig. 3, the Payload field of an REQ message consists of four sub-fields. The Passenger ID field records the identification of passenger's handheld device or other unique identification. The Passenger position field records the location where the passenger broadcast the REQ message. The Number of taxis field records the number of taxis demanded by the passenger, and the TTL field records the remaining number of hops that the message is required to forward. After broadcasting REQ message, passengers wait for a specific time. If a passenger cannot receive any REP from taxis during the specific time, he will increase the value of TTL by two for each increasing and send the REP, again.

Passenger ID	Passenger position	Number of taxis	TTL
4 bytes	8 bytes	1 byte	1 byte

Figure 3. The *Payload* field of Request message (REQ).

When broadcasting messages in VANETs, in order to reduce communication overhead, the receiver who is closest to destination is responsible for forwarding the message [8]. Receivers wait for a back-off time before forwarding REQ messages. Receivers who are closer to destination have shorter back-off time and could send messages early. When receiving the same REQ before the back-off time $W_b(dist)$ is expired, receivers drop received REQ messages. The back-off time, $W_b(dist)$ is defined as follows:

$$W_b(dist) = \left[\frac{(Tx\_Range - dist)}{l} \times \alpha + rand\_number\right] \times Time\_Slot, \quad (1)$$

where *dist* denotes the distance from the receiver to the sender, *Tx_Range* denotes the largest communication radius of wireless network (it is 250 meters in simulations), *l* (it is 50 meters in simulations) and *α* (it is 10 in simulations) are two parameters, *rand_number* is a random number from 0 to 9. *Time_Slot* denotes the duration from broadcasting a message to receiving it by others (it is set 1 ms in simulations).

Taxis changes from vacancy mode to contention mode when receiving REQ from passengers. According to traffic regulation like one-way traffic, prohibition of left turn or prohibition of U turn that are provided by the electronic maps, each taxi is able to calculate the driving path and driving distance from present position to passengers. The determined driving distance is recorded in the Payload field of REP message as shown in Fig. 4. And the next intersection on the determined driving path is recorded in the Destination position field as shown in Fig. 2. Then the REP is forwarded for competing for passengers.

Taxi ID	Taxi position	Passenger ID	Passenger position	Number of taxis	Driving distance	Last intersection position
4 bytes	8 bytes	4 bytes	8 bytes	1 byte	2 bytes	8 bytes

Figure 4. The *Payload* field of Reply message (REP).

In Fig. 4, the Payload field of REP message consists of seven sub-fields. The Taxi ID field records the identification of taxi. The Taxi position field records the present position of taxi. The Driving distance field records the distance from taxi's present position to passenger's position, and the Last intersection position field records the position of last intersection that the message passed by. The definition of remaining three sub-fields is the same as that of Fig. 3.

Recall that REP messages are transmitted along the determined driving paths. Hence, taxis/forwarders put next intersection on their determined driving paths in Destination position field (see Fig. 2). Next intersection on the driving path can be determined by the aid of the driving direction and the current position of taxis/forwarders. When a vehicle/taxi receives the REP message, it is called a receiver and competes for becoming a forwarder. Receivers calculate their back-off time and start to forward the REP message when back-off time is expired. Receivers drop received REP messages when receiving the same REP message before their back-off time is expired. Their back-off times are determined as follows.

Case 1. Receivers are at or near the destination (records in the Destination field). If the distance from receiver to destination is short enough, e.g., less than d meters, then receiver is regarded as arriving at the destination and could forward the REP message to next destination (intersection). In such case, back-off time $W_s(dist)$ of receiver is defined in formula (2). Otherwise, the procedure will be described in the next paragraph.

$$W_s(dist) = \left\lceil \frac{dist}{l_s} \times \alpha + rand\_number \right\rceil \times Time\_slot, \qquad (2)$$

where *dist* denotes the distance from receiver to destination (recorded in the Destination position field) and l_s is a constant (we set 5 meters in simulations). Receivers modifies the REP messages they received: set the Destination position field to be the determined next intersection on the driving path and the Last intersection position field to be the current intersection (i.e., their current position). For example, in Fig. 1, if taxi *B* wants to compete for passenger *P*, it sends an REP message to intersection 1 which is the first intersection on the driving path to passenger P. While the REP is sent to intersection 1, vehicles/taxis that receive this REP will calculate next intersection by the aid of Last intersection position field of REP and their current position. Considering the traffic regulation, the next destination is intersection 2 which is on the shortest driving path to the passenger. The REP will be sent out by the receiver whose back-off time $W_s(dist)$ is expired first. If the sending task is successful, then the winner becomes a keeper, which keeps the taxi's REP information at this intersection, to reduce the blocking time (the detail can be found in Section 3.2). Other receivers give up forwarding when receiving the same REP.

Case 2. Receivers are not at or near the destination. Suppose that receivers is not near or at the destination (i.e., its distance to the destination position is larger than d meters), but closer to the destination than sender's. Then the back-off time $W_f(dist)$ of receiver could be calculated by the following formula:

$$W_f(dist) = W_b(dist) + Max\_W_s, \qquad (3)$$

$$Max\_W_s = \left\lceil \frac{d}{l_s} \times \alpha + max\_rand \right\rceil \times Time\_Slot. \qquad (4)$$

where *dist* denotes the distance from the receiver to the sender. Obviously, receivers at or near the destination should have higher priority to forward the REP message. So we let the *max_rand* is the maximum value of rand-number defined in formula (2).

REP messages are transmitted along the shortest driving paths of taxis under traffic regulation orders. REP message acts like an agent of taxi. It travels from taxi's current position to the passengers. The travel time of REP message gives an image of driving distance of taxi, i.e., the travel time of REP message increases as the driving distance of the taxi increases. Hence, the taxi of the REP first received by the passenger is regarded as the taxi with the shortest driving distance. Another advantage of transmitting REP along the shortest driving paths of taxi is that a vacant taxi with the shortest driving distance could be booked without prolonging the waiting time of passengers (to collect taxis' REP).

B. Taxi De-blocking

Vacant taxis which attempt to compete for the passengers are in contention mode after sending REP. In order to guarantee that no two passengers book the same taxi simultaneously, taxis in contention mode are blocked from joining competitions for other passengers. In [1], taxis in contention mode are aware of their failure if they have pended for more than predefined maximum blocking time and do not receive ACK from the passenger. In our paper, the maximum blocking time w is defined as follows:

$$w = \frac{Driving\_distance}{Tx\_Range/2} \times Max\_Hop\_Delay \times 2, \qquad (5)$$

where *Driving_distance* denotes the driving distance to the passenger. Notice that, the maximum number of transmissions (of an REP message) is *2x/Tx_Range*, where x denotes the driving distance. This is because, for three successive forwarders, the minimum distance between the first forwarder and the third forwarder is greater than the transmission radius *Tx_Range*. *Max_Hop_Delay* denotes the maximum delay time per hop/transmission, which is defined as follows.

$$Max\_Hop\_Delay = \left(\frac{(Tx\_Range/2)}{l} \times \alpha + max\_rand \right) \times Time\_slot + Max\_W_s \qquad (6)$$

where *max_rand* is the largest random value defined in formula (1), and Max_W_s denotes the maximum value of W_s defined in formula (2).

In order to further reduce blocking time of taxis, a staying message (STA) is left at intersections which an REP passes through. The STA messages are used to keep the driving distances from vacant taxis to passengers. By the aid of STA, taxis are able to know in advance whether there is another taxi with shorter driving distance competing for the same passenger. Let the vehicle who keeps the STA messages at an intersection be the "keeper". By comparing the driving distances recorded in REP and STA, the taxi with longer driving distance fail the competition, and the keeper would send a reject message (REJ) to inform the taxi of its failure. Taxis fail the competition are in vacancy mode again and are able to compete for other passengers. The STA messages help taxis with longer driving distance to be aware of their failure early. Therefore, the average blocking time could be reduced.

The format of STA is very similar with REP (see Fig. 5). This is because the main function of STA is to leave the track of REPs which have passed through intersections.

Taxi ID	Taxi position	Moving distance	Position of intersection	Passenger ID	Number of taxis	Life time
4 bytes	8 bytes	2 bytes	8 bytes	4 bytes	1 byte	4 bytes

Figure 5. The *Payload* of Staying message (STA).

The definition of the Taxi ID field, the Taxi position field, the Passenger ID field and the Moving distance field are the same with the definitions in Fig. 4. The Position of intersection field is the position of intersection where the STA stays. Number of taxis field is the number of taxis demanded by passengers. If the value of Number of taxis is not zero, it means that the passenger needs more than one taxi, and the value decreases while REPs of different taxis have passed through. If the value of Number of taxis is zero, REPs with longer driving distances are prohibited to pass through (and REJ messages are sent back to taxis of these REPs).

Notice that, the life time of STA staying at an intersection close to the passenger is longer than that of STA which is far apart from the passenger. The life time of STA is defined as follows:

$$Life\_time = \frac{Max\_Dist - dist_p}{Tx\_Range/2} \times Max\_Hop\_Delay \quad (7)$$

where $dist_p$ denotes the remaining driving distance to the passenger's position. *Max_Dist* (it is 2 kilometers in simulations) denotes the farthest driving distance of taxis that could receive the requests of passengers. Recall that *Tx_Range/2* denotes the average distance between a forwarder and its successive forwarder. In order to guarantee that STA lives when REPs pass through the intersection to competing for the passenger recorded in STA, the life time of STA is the time of transmitting REPs of taxis which is farthest to the intersection.

For an intersection, at least one road passing through it is at red light, i.e., there are vehicles waiting for the red light. In our protocol, STA is kept by those taxis/vehicles which are waiting for the red light. When the keeper of the STA receives an REP which competes for the same passenger, then an REJ will be sent to the taxi of REP if the driving distance recorded in REP is longer than that of STA. Otherwise, the original STA terminate its life and the REP leaves a new STA. For example, in Fig. 3.6, the REP of taxi *A* passed through intersection 1 first and leaved a STA. After a while, the REP of taxi *B* arrives at this intersection and is compared with the STA of taxi *A*. Since left turn is prohibited at intersection 1, the driving distance of taxi A is longer than that of taxi B although REP of taxi A passes through intersection 1 first. So only the STA of taxi B is left at intersection 1. Then keeper transmits a reject message (REJ) to taxi A by using the position-based forwarding scheme.

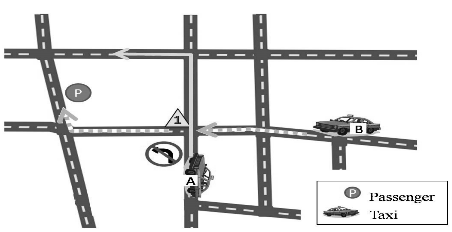

Figure 6. REP of taxi *A* passed through intersection 1 first.

The REJ message includes the Taxi ID field, the Passenger ID field and the Passenger position field. Taxis who receive the REJ message return to the vacancy mode. When other vehicles received the REJ, they stop to transmit the REP whose Taxi ID field is equal to the Taxi ID field of REJ. Hence, communication overhead could be reduced by stop transmitting REPs which fail in the contention.

If the life time of STA is not expired when the keeper will leave the intersection, the keeper will find another vehicle near the intersection to be a successive keeper. First, the keeper broadcast the STA message, and then other vehicles which receive the message calculates their own back-off time $W_k(dist, speed)$. Clearly, vehicles which are the closest to the intersection and have the slowest speed are proper to be new keepers. So, we define $W_k(dist, speed)$ as follows.

$$W_k(dist, speed) = \left[\left(\frac{dist}{l_s} \times \alpha + rand\_number\right) + speed\right] \times Time\_Slot, \quad (8)$$

where l_s and *rand_number* are the same as that of formula (2). *dist* denotes the distance from the receiver to the intersection which STA stays. Speed denotes the driving speed (km/hr) of receivers. A receiver whose back-off time has expired transmits a confirmation message to the original keeper and becomes a new keeper. The mission of the original keeper is terminated when it receives the confirmation from the new keeper. If the original keeper has not received any confirmation from other vehicles, it has to continuously broadcast STA. Other vehicles competing for becoming keeper give up when receiving the confirmation of the new keeper.

STAs is used as a track of REP that passed through the intersections. STAs stop the transmission of REPs which correspond to taxis with longer driving distances, which can reduce the amount of REP messages. Besides, REJ is used to de-block failed taxis early. So, the blocking time of taxis could be reduced. This scheme reduces not only the blocking time of taxis, but also the possibility of congestion of wireless networks (especially, near by the passenger).

Passengers send ACKs to book those taxis whose REPs arrive at passengers first. The ACK message includes the Taxi ID field, the Passenger ID field and the Passenger position field. Of course, booked taxis should drive to pick up passengers.

III. SIMULATION RESULTS

In this section, the performance of our distributed taxi hailing protocol and the probabilistic on-demand protocol of EZCab [1] are compared. When a passenger wants to hail a vacant taxi, EZCab broadcasts request messages and builds a communication tree *T*. A vehicle close to the passenger is chosen as the root of *T*. In *T*, nodes denote taxis. Two nodes are connected by a tree edge in *T* if they are communicable with each other. Each node reports it knowledge about vacant taxis to its parent nodes. And then root chooses the vacant taxi that has the less hop counts to it (i.e., the less hop counts to the passenger) to pick up the passenger.

We use VanetMobiSim/ns-2 [2] since it offers the possibility of specifying realistic mobility models, using VanetMobiSim, and communication environments, using ns-2. In our simulation, there are 200 taxis with random destinations and speed 0 to 60 km/hr. Traffic lights change every 60 seconds. The communication range of wireless communication equipments installed in taxis or passengers' handheld devices is assumed to be 250 meters. We set ten vertical roads and six horizontal roads in a square range of 2000 meters × 3000 meters.

In the simulations, the density of vacant taxis (i.e., number of vacant taxis / total number of taxis) varies from 10% to 60% and the result is obtained from the average results of 20 simulations. We analyze the booking time, the driving distance of booked taxis driving to passengers, blocking time of taxis, and the number of transmitted packets.

Booking time vs. density of vacant taxis. In our simulation, booking time is defined to be average execution time of the given scheme (i.e., duration from request message broadcasted by passenger to confirmation message received by the booked taxi).

Fig. 7 shows booking time decreases as density of vacant taxis increases. The ratio of booking time of our protocol to that of EZCab is not greater than 60%. With the increasing density of vacant taxis, the ratio decreases. This is because the possibility that there is a vacant taxis with very short driving distance to passenger increases as density of vacant taxis increases. Although reply messages in our protocol are transmitted along the driving paths from taxis to passengers instead of the shortest paths, but it is not wasting too much time in our simulation results. On the other hand, EZCab has to waste time on building a tree and acquire information of vacant taxis by searching the whole tree.

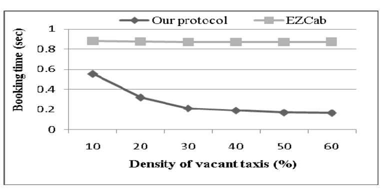

Figure 7. Booking time versus density of vacant taxis.

Booking time vs. number of requests. In this experiment, each passenger is assumed to hail only one taxi and the density of vacant taxis is assumed to be 10%. Fig. 8 shows that booking time of our protocol increase little with the increasing number of requests. This is because the density of vacant taxis decreases after other passengers hailed taxis successful. Fig. 8 also shows that booking time per request of EZCab is the same no matter how many requests there are. This is because that the time of building the tree topologies and acquire information of vacant taxis by searching the whole tree is the same.

Driving distance vs. density of vacant taxis. Fig. 9 shows that there is significant reduction on driving distance from booked taxis to the passengers in our protocol. The driving distance in our protocol is not greater than 50% of that in EZCab. It also shows that in most cases, the driving distance monotonically decreases with the increasing density of vacant taxis. This is because, with the increasing density of vacant taxis, there is higher possibility to have vacant taxis close to the passenger. However, considering the actual driving directions and traffic regulations, closer vacant taxi may not have shorter driving distance. Hence, curves in Fig. 8 are not decreasing. On the other hand, since booked taxis in EZCab are closest to passengers in tree topology, the driving distance of booked taxis in EZCab is longer than ours.

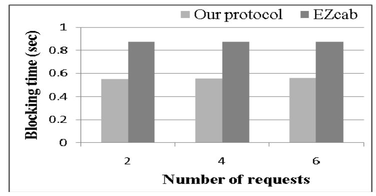

Figure 8. Booking time versus number of passengers.

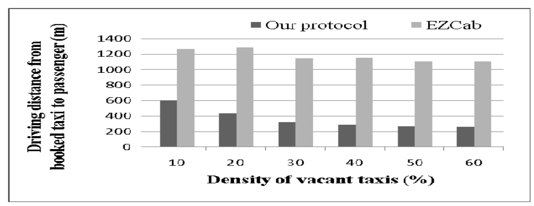

Figure 9. Driving distance from booked taxis to passengers versus density of vacant taxis.

References

[1] P. Zhou, T. Nadeem, P. Kang, C. Borcea and L. Iftode, "EZCab: A Cab Booking Application Using Short-Range Wireless Communication," in *Proceedings of IEEE Pervasive Computing and Communications*, pp. 27-38, Mar. 2005.

[2] The Vehicular Mobility Simulator, http://vanet.eurecom.fr/

[3] J. Zhao and G.Cao, "VADD: Vehicle-Assisted Data Delivery in Vehicular Ad Hoc Networks," in *Proceedings of IEEE Computer Communications*, pp. 1-12, Apr. 2006.

[4] Z. Liao, "Real-Time Taxi Dispatching Using Global Positioning Systems," *Communications of the ACM*, vol. 46, pp. 81-83, May. 2003.

[5] T. Nadeem, P. Shankar, and L. Iftode, "A Comparative Study of Data Dissemination Models for VANETs," in *Proceedings of ACM Mobile and Ubiquitous Systems: Networking & Services*, pp. 1-10, Jul. 2006.

[6] P. I. Philippopoulos, G. Soulos, A. Krukowski, N. Argyreas, V. Gorgas, D. Kyriazanos, D. I. Axiotis and S. C. A. Thomopoulos, "ToD: An Intelligent Location-Based Transport Management System," in *Proceedings of Intelligent Vehicles Symposium*, pp. 793-798, Jun. 2007.

[7] H. Wu and R. Fujimoto, R. Guensler, and M. Hunter, "MDDV: A Mobility-Centric Data Dissemination Algorithm for Vehicular Networks" in *Proceedings of ACM Vehicular ad hoc networks*, pp. 47-56, Oct. 2004.