Hindawi Publishing Corporation
The Scientific World Journal
Volume 2014, Article ID 601913, 9 pages
http://dx.doi.org/10.1155/2014/601913

Research Article

A Lightweight Neighbor-Info-Based Routing Protocol for No-Base-Station Taxi-Call System

Xudong Zhu, Jinhang Wang, and Yunchao Chen

School of Computer Science & Information Engineering, Zhejiang Gongshang University, Hangzhou 310018, China

Correspondence should be addressed to Xudong Zhu; zhuxd@zjgsu.edu.cn

Received 26 October 2013; Accepted 28 November 2013; Published 9 March 2014

Academic Editors: W. Sun, G. Zhang, and J. Zhou

Since the quick topology change and short connection duration, the VANET has had unstable routing and wireless signal quality. This paper proposes a kind of lightweight routing protocol-LNIB for call system without base station, which is applicable to the urban taxis. LNIB maintains and predicts neighbor information dynamically, thus finding the reliable path between the source and the target. This paper describes the protocol in detail and evaluates the performance of this protocol by simulating under different nodes density and speed. The result of evaluation shows that the performance of LNIB is better than AODV which is a classic protocol in taxi-call scene.

1. Introduction

The VANET has characteristics of mobile ad hoc network, such as self-organization, flexible structure, multihop routing, dynamic changes of network topology, and need of good scalability. Special application environment, such as narrow road, high density node distribution, high-speed node moving, and other factors, will directly influence the information transmission ability of VANET network, which leads to the increase of packet loss and delay. Meanwhile, the limited wireless channel and huge network scale challenge the data transmission of VANET.

The key to solve the above problems is how to design a routing protocol [1–7]. Since the nodes in VANET are of high mobility, it results in connection failure while applying routing protocol of the traditional MANET [8–10], such as AODV [11–13] and DSR [14, 15], into the VANET, thus leading to delay or even failure to transmit information to the target node.

Due to the special environment in VANET, it cannot have a widely applicable routing algorithm. The typical algorithm in VANET, such as GSR routing protocol, involves each node storing the neighbor lists, topology table, next hop table, and distance table. All of them maintain the state information of adjacent nodes and choose the appropriate router according to the location and topological information. In small network with high mobility and limited bandwidth, the performance of transmission is good However, it requires the node to maintain the network topology. And with the increase of network size, the routing information that needs to be exchanged will increase exponentially. Another type of algorithm, such as the GPSR [16, 17], is based on position. It depends on the overall geographical location information search system. It cannot work without GPS.

When applied to the urban taxi-calling scene, the VANET vehicle network has the following characteristics. (1) The density distribution of vehicle is uneven. Places such as the business center, theaters, and bus stations are dense. The traditional VANET routing protocols have serious information channel congestion problem. (2) The calling passenger has no special requirements to the not-taken taxis, so all the nearby taxis can serve. While the traditional VANET routing protocols always have designated transmission target, or specifies the routing service of all nodes in the target area, which will lead to huge routing overhead, further deteriorating the channel congestion problem. (3) The data transmission is time valid. Passengers' calls need to be responded to in a short time, while the routing connection time is fairly long in some of the traditional VANET routing protocols. Therefore, it cannot effectively find not-taken taxis with such application and will have bad effects on passengers' experiences. We describe some of the work roughly in [18].

This paper aims to solve the quick topology change, short connection duration, and unstable wireless signal quality of the VANET network, thus proposing a kind of lightweight routing protocol applicable to the urban taxi-call without base stations. The protocol maintains and predicts neighbor information dynamically in order to find the reliable path between the source and the target. This paper describes the protocol in detail and evaluates the performance of this protocol by simulating under different nodes density and speed. The result of evaluation shows that the performance of LNIB is better than that of AODV which is a classic protocol in taxi-call scene.

The second chapter analyzes the application and proposes the model; the third chapter describes the routing protocol and its algorithm; the fourth chapter is about the experiment and data analyses; chapter five concerns other related research; the last part will draw a conclusion.

2. Transmission Mode and Analyses

In the taxi-calling scene, the vehicle network has two sets: taxi nodes $Q = \{q_1, q_2, \dots q_n\}$ and passenger nodes $P = \{p_1, p_2, \dots p_m\}$. From the start of the call to get on the car or the call to cancel, a connection should be set up in the network. Because the passenger always waits at the original place or moves slowly, suppose p is a static node. And p only acts as the initiator of data transmission or receiver, not as mediator in other routings. The node q can send, receive or act as a routing node to transmit data. Assume that the speed scalar of each node in Q is $|v|$.

When p initiates a call, the message transits through the taxi nodes, so p can quickly find nearby not-taken cars node q_s that can meet the requirements.

Supposing there are two nodes a and b in the network, the signal coverage radii for them are, respectively, R_a and R_b. If the distance between a and b is $D_{a,b} \le \min(R_a, R_b)$, it can be called "meet." To be more general, assuming that signal coverage radius of each node in the network is R, then "meet" in time t means $D_{a,b}(t) \le R$. Define H_{a,q_s} as the minimal number of hops to transmit message from a to q_s. Among all q_s that get the call request, H_a is the lowest of H_{a,q_s}.

2.1. Prediction Based on Neighbor Information. In order to percept the current meet nodes in real time, q needs to send broadcast packets to notice their own state information periodically. In the starting time t of every broadcast cycle, q measures their own motion state, including the speed, the current position, and other information. According to the above information, q predicts the coordinates at the time of $t + t_f$ and broadcasts the position. The broadcast cycle is t_b. If t_b is small, then the channel will be occupied by a large number of broadcast packets, thus influencing the normal data transmission. However, large t_b will lose the prediction value due to the frequent changes in actual situation. Therefore, t_b should be set properly, such as 10 seconds. A broadcast cycle can be divided into multiple correction cycles. The node in a correction period measures their movement state. While finding the coordinates' different Δs between the current forecast coordinate at $t + t_f$ and the last broadcast coordinates surpasses, it indicates that the vehicle's driving condition changes. If Δs is greater than a certain value, the node broadcasts the position again to correct the predict position of itself in neighbor nodes' state table, thus to ensure the prediction accuracy. Suppose the correct period is t_r.

Meanwhile, if q receives the position broadcast of the nearby nodes, it maintains an adjacent link (see Table 1) to record node information that meets with q. It can be seen from the table that q_2 measures its position and velocity at 13:11:12$'$36 and broadcasts. q_1 receives the position broadcast. If $D_{q_1,q_2}(t) \le R$, then it can be predicted that, at $t + t_f$, q_1 and q_2 will keep meeting. If q_2 is not in the table, q_1 adds the information of q_2 to the table and sets q_2 to be active. If q_2 is already in the table, update the information of q_2. When q_1 receives the correction broadcast of q_2, the position information of q_2 also needs updating. If it can be predicted that q_1 and q_3 will not meet at $t + t_f$, set q_3 to be inactive. As to the inactive node in the table, if no broadcasts are received in more than one broadcast cycle, then delete the entry of this inactive node. Each time the passenger calls a taxi, several links will be produced. If q_1 and q_3 are neighbors and are both in one link, then they set up link ID in the entry for each other in their own table. Link ID is a large number that is randomly generated. It can be assumed that it is unique in a limited region.

When q_1 receives the broadcast of neighbors at time t, it will predict coordinates of itself and neighbor nodes at $t + t_f$, respectively. If the two cars can keep communication at that time (such as q_2), the communication is effective. Therefore q_1 stores the information of q_2 in the table otherwise discards it (such as the q_3).

In order to provide the position information and routing of not-taken taxis, the hop distance should be introduced when q node is broadcasting message (including position and correction broadcast). Periodically q_s sets H_{q_s} to 0 and sends broadcast. The q_s discards any broadcast information it received without processing. The q sets H_q value to ∞, when no neighbor nodes can reach q_s node. When q_1 receives the broadcast of q_2 with the active state, if $H_{q_2} + 1 < H_{q_1}$, q_1 modifies the H_{q_1} value to $H_{q_2} + 1$.

Once H_{q_1} changes, q_1 will spread to notify neighbor nodes to be updated by periodical breakfast. But each cycle can only update 1 hop on the link. If $H_{q_1,q_2} = n$, it needs n cycle to update H_{q_2}.

In order to improve the broadcast speed of H, the sending phase of each cycle is divided into two time slices. The node of H changes, but no broadcast will be broadcasted at the first time slice. If the H value has no change after last broadcast, the node broadcasts in the second time slice. Thus, the value of H may spread 2 hops in a cycle. For the length of link will be limited in 3-4 hops in the actual system, two cycles can complete the information disclosure of not-taken taxis by the improved broadcast speed.

2.2. Links Setting and Handshake. When p launches the call, it needs to find suitable not-taken taxis q_s as soon as possible and make an appointment. In this process, the following

Table 1: State of the q_1's Neighbor Nodes.

Neighbor	H_{q_n,q_s}	Position	Direction of speed	Link ID	Reliability	Measuring time	State
q_2	3	120.213476E: 30.321768N	d_{q_2}	182378627	0.94	13:11:12′36	Active
q_3	5	120.213484E: 30.321775N	d_{q_3}	452766732	0.83	13:10:37′34	Inactive

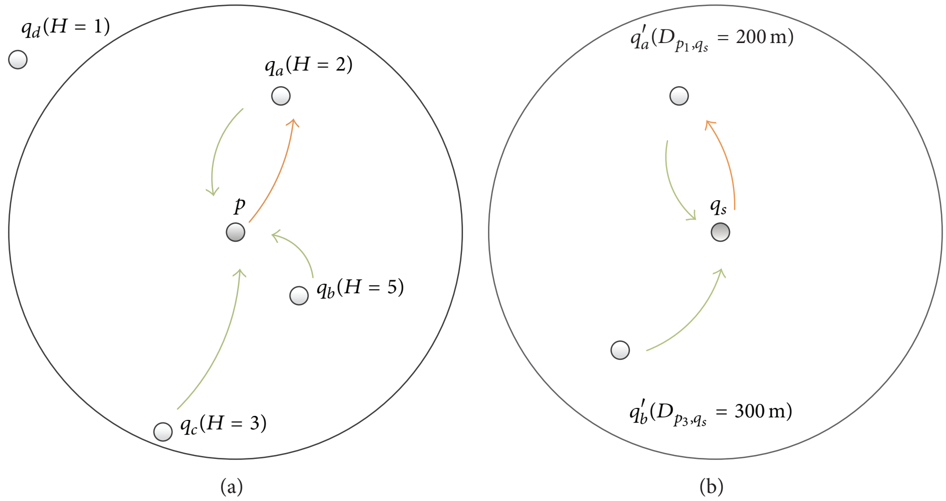

$q_d(H = 1)$
$q_a(H = 2)$
p
$q_b(H = 5)$
$q_c(H = 3)$
(a)
$q'_a(D_{p_1,q_s} = 200\,\mathrm{m})$
q_s
$q'_b(D_{p_3,q_s} = 300\,\mathrm{m})$
(b)

Figure 1: (a) If there are the q nodes within the single hop range of p, then they shake hands and choose one as q_a that has minimal hop distance to q_s to make an appointment. (b) If there are several q nodes around q_s and several p nodes, send the appointment requests, then they shake hands and choose the nearest p node to response.

questions need to be solved. First, how to find q_s in the shortest time and establish the link to q_s? Secondly, if there are several q_s nodes, how to choose one of the most suitable ones to make an appointment?

The simplest method is to use the broadcast to send out the call until a node of q_s receives the request and returns a response. The response will go backtracking to the p node. If p receives several responses, then choose one from them, after which appointment confirmation will be sent to the chosen node through the path just set up. Meanwhile, it will send negative responses to other not-taken taxis. This method is easy to be implemented, but in the real VANET scene, there are serious problems; the message must pass the whole link three times: request, response, and appointment. On the one hand, it will lead to serious delay and difficulties in keeping the link; on the other hand, the flooding of breakfast will deteriorate the network environment and influence the normal communication.

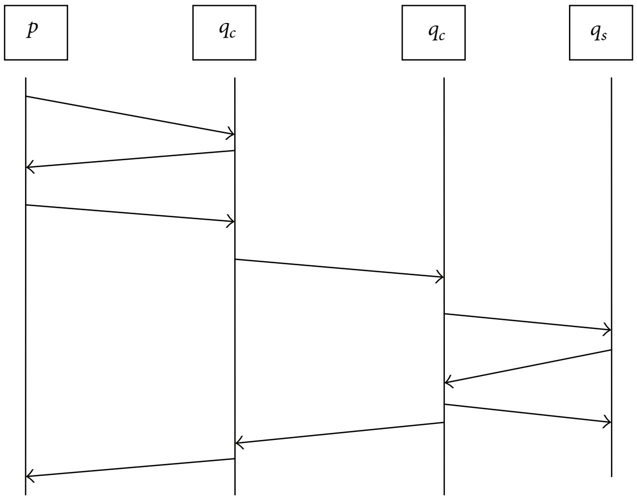


Figure 2: The two-end handshake agreement.

This paper proposes a two-end handshake agreement to great link (show as Figures 1 and 2) According to the neighbor nodes' information described in Section 2.1, each node can know the nearest (by hop distance) q_s and the shortest route to arrive at the node. If p sending the request broadcast, its neighbor q node which receives p's breakfast directly can judge whether it can reach q_s or not. In fact, it is supposed that the whole data transmission is a single route (the node on the link will transmit a message to only one neighbor node). Once q is identified as the first hop, the routing from p to q_s will also be established. The q can directly send a request response to the p. If p receives responses from several q nodes, it chooses a q node as q_a to make appointment confirmation and send a veto appointment to other response q. After that q_a accepts the confirmation and sends it to the q_s

based on the routing information, that is, to transmit message to the neighbor q node in which $H_q = H_{q_a} + 1$. On the routing, node q' will record the information of last hop, while receiving the message. Meanwhile, transmit to the next hop which $H_q = H_{q'} + 1$ until sent to the node which $H_q = 0$, that is q_s.

The q_s end of the link also needs handshaking. If several neighbor q nodes, in which $H_q = 1$, are for the same q_s node, they will send appointment requests to the q_s, and the q_s sends a confirmation response to one of them and negative responses to others.

The two-end handshaking method can reduce the three times handshaking to two times, thus reducing the risk of delaying and disconnection. Meanwhile, the broadcast can be controlled in 1 hop and can avoid the flooding.

2.3. Disconnection. Suppose $H_{q_s} = 0$, $H_{q_1} = 1$, $H_{q_2} = 2$, and $H_{q_3} = 3$. If q_2 receives the position correction information from q_1, the adjusted position predictions indicate that q_1 will lose contact with q_2. Since H_{q_1} is the minimum H stored in q_2, after q_2 node deletes the q_1, it modifies its own H value to ∞ and triggers an extra position correction broadcast. Also, when q_3 receives the position correction by this broadcast, it continues the same operation.

In the communication process, the effectiveness of link should be guaranteed from the sending of p's request to q_s's response return to p. Once a pair of nodes on the link are disconnected, it should be processed correspondently. The relatively ideal way is to find the third node before disconnection and to maintain the link. Another simple method is to notify p and q_s to abandon the appointment on both ends. Thus, it can avoid feedback loss which will result in information consistency. Then the p initiates the request process again.

As the vehicle's speed is fast and unstable, the time of link maintains a relatively short time. The frequently broken links will lead to repeated reconnection. To avoid this, higher reliable links are required. So in the q's neighbor nodes state table, the reliability attribute is added for each neighbor node. The reliability refers to the prediction accuracy at $t + t_f$ and the help to effective datagram transmission on the link. It mainly depends on the following two conditions: (1) during t to $t + t_f$ period, the relative position between q' and q. If the distance is far, on the one hand, the weak signal can lead to transmission instability; on the other hand, since q' has been close to the edge of the q communication range, the slight change of the vehicles will make q' out of q's communication range and cause broken connection. If q' and q are close, the data transmission will take place in a short distance. The datagram will not move a lot in geographical location; thus, it influences the efficiency of data transmission. Thus supposing $U_{q,q'}(t') = 2|D_{q,q'}(t') - R|/R$ refers to the transmission reliability of q, and q'

$$U_{q,q'}\left(t, t + t_f\right) = \frac{\text{integration}}{\left(t, t + t_f\right)}. \quad (1)$$

Modify the condition of correcting H_q in Section 2.1. Given the reliable threshold $r \in [0, 1]$ and after q_1 node receives the broadcast of q_2 node, if state (q_2) = active, $H_{q_2} + 1 < H_{q_1}$, and $U_{q,q'}(t, t + t_f) > r$, then modify the H_{q_1} value to be $q_2 + 1$.

3. Protocol Implementation

In order to further introduce the implementation of the protocol, the pseudocode algorithms of taxi q node and passenger p node are given, respectively, in this section. To be simple, based on the discussion in the second section, the following several types of datagram in the taxi-call network will be referred to.

Broadcast (L, V, H, t_c) represents position broadcast. The taxi node sends periodically, and the nearby taxi node receives.

M_{1-6} represents the six kinds of datagrams in the two-end handshaking protocol after the passengers make a request.

M_e represents the cancelation of the reserved datagram while finding the abnormal disconnection.

Algorithm 1 is divided into two parts. The timer gets the position and speed information for a short time (e.g., 3 seconds). If the interval exceeds t_f, then send broadcast to the nodes around. According to the current measured data, the position at $t + t_f$ can be predicted. If there exists significant difference between the position and the position of the last broadcast, then send broadcast without considering whether the time intervals exceed t_f or not. The timer will also inspect the neighbor node that is not updated over time in the adjacency list and set these nodes as inactive. Since the time in the record is the position measuring time, to consider the communication delay, the overtime window will be slightly amplified to $1.1t_f$. If not updated for more than two broadcast cycles, then delete the node.

As for the inactive nodes, check whether they are in the current communication connection. If it is true, then send M_e to other connected nodes to cancel the conformation.

The other part of Algorithm 1 is the listener. After the listener receives a datagram, he/she judges data types. If it is the position broadcast datagram, then update the list. Once the predicted node position changes are identified in the current connection and in a short time will overpass the scope of communication, and then notify other nodes on the connection to cancel the appointment. When accepting other types of datagram, again transmit or respond based on the protocol.

Algorithm 2 mainly includes two timers and a listener. When initiating the request, timer$_1$ is set. At the same time, the listener will put the received response to the response queue. When timer$_1$ is triggered, it checks the queue. If the queue is empty, it means no response, and it sends the request again. Otherwise, it chooses the response that has the nearest transmission distance from passengers to not-taken taxis in order to make appointment. And it simultaneously sends veto to other response nodes. After sending the appointment datagram, set timer$_2$. Once no appointment conformation M_4 in a certain time M_4, then the timer$_2$ will be launched, and send request M_1 again. Attention here, once the listener receives an M_e, the datagram cancellation is caused by

```
**Initialization** $t = 0$;
**Initiate the prediction position** $S = 0$;
**Initialization** $H = \infty$;
Set timer;
Timer_task(){
    Obtain the current position $L$, speed $V$, time $T_c$
    $H_{\text{new}} = H$;
    If (no-load){
        $H_{\text{new}} = 0$;
    }
    Calculate prediction position $S'$ at $(t + t_f)$
    If ($t_c - t \geq t_f$){
        $H = H_{\text{new}}$
        broadcast($L, V, H, t_c$);
        $t = t_c$;
        $S = S'$
    }else if ($|S - S'|$ larger than threshold || $H - H_{\text{new}} \langle\rangle 0$){
    $H = H_{\text{new}}$
        broadcast($L, V, H, t_c$);
        $t = t_c$;
        $S = S'$;
    }
    For each node $q$ in table{
        get $q$'s prediction time $t_q$
        if (state($q$) == active && $t_c - t_q \geq 1.1t_f$){
            state($q$) = inactive;
            if ($q$ have the link ID){
              send $M_e$ to other node which have the same lnik ID as $q$;
            }
        }
        If ($t_c - t_q \geq 2t_f$){
              Remove $q$ from table;
         }
    }
    Update $H$;
}
Count = 0; // reset 0 after passengers getting off or appointment cancellation
Block_listen(Message){
      Switch typeof(Message) {
         Case $M_b$:
              update table
              update $H$
              if $q$ node that sends $M_b$ gets out of the communication range, and it has link ID in the table
                     send $M_e$ to other node
                     which have the same link ID as $q$;
              break;
         Case $M_1$:
              If ($H < \infty$){send $M_2$ back}
              break;
         Case $M_3$:
              If (Message == Reserve){
                   If ($H$ == 0 && count == 0){
                          Count = 1;
                          send $M_4$(Agree or Disagree) back;
                   }else if ($H$ == 1){
                        Set link ID;
                        send $M_3$ to all $q$ in table which $H_q = 0$;
                    }else{
```

Algorithm 1: Continued.

```
                        Set link ID;
                        send $M_3$ to one $q$ which $H_q = H - 1$;
                    }
                }
            Case $M_4$:
                If (Message == Agree){
                    Send $M_4$ forward to $q$ which $H_q = H + 1$
                        and have the same link ID;
                    delete link ID;
                }
                If ($H$ == 1){
                    Send Confirm back;
            }
            Default:
        }
    }
```

Algorithm 1: Algorithm of taxi node.

```
Initiation queue = empty
Call_taxi{
    Get its position $L$
    Broadcast($M_1$)
    Set Timer$_1$();
}
Timer1_task(){
    If (queue = empty){
        Call_taxi();
    }else{
        Choose the nearer hop $q$,
        Produce link ID;
        Send $M_3$(Reserve, link ID) to $q$;
        Send $M_3$(Negative) to other
    }
    Set timer$_2$;
}
Timer$_2$_task(){  Call_taxi(); }
Block_listen(Message){
  Switch typeof(Message) {
      Case $M_2$
          Add $q$ to queue;
          Break;
      Case $M_4$:
          Cancel Timer$_2$;
          Break;
      Case $M_e$:
          Set Timer$_2$ current time;
          Break;
      Default:
}
```

Algorithm 2: Algorithm of passenger node.

connection error. Then immediately start the timer$_2$ and send request again.

4. Performance Analysis

In order to evaluate the performance of the proposed routing protocol, the paper uses NS2 and VanetMobiSim to simulate.

The NS2's main parameters are shown in Table 1. The simulation scene size is 5 × 5 square kilometers. In VanetMobiSim simulation, the scene size is 5 × 5 square kilometers too. In the total area there are 20 traffic signals and the entire simulation area is divided into several regions. In every region the density of streets and the allowed maximum speed are different. The regions which have higher street density have lower allowed maximum speed. When vehicles get into the corresponding region, they choose the speed from street allowed maximum speed and self-allowed maximum speed with a low one. In all of our simulations, the empty taxis account for 40%.

We set the interval of route maintain message T_b to be equal to 10 s and then set the maximum speed of taxi nodes to be equal to 60 km/h. When we adjusted the density of taxi nodes, we got the probability of passenger node P to find empty taxi node T_D and the probability of node P to receive the message from T_D as shown in Table 3.

As shown in Table 3, when node density is too low, the transmission range will be so small that it is hard to maintain the information of nodes for a long time. Hence, when P appears, the number of nodes that P can communicate with is small and the time of link is short so that it is hard to communicate with T_D. As the node density becomes higher, situations are getting better, but some nodes have higher velocity, which leads to more failure. Because the links chosen are the links that have long duration, so the probability of node P receiving acknowledgement messages from node T_D is pretty high.

Then if we fix the density of taxi nodes as 5 per km^2 and fix the transmission range as 500 m, we will get the relation of the maximum speed and the successful ratio.

We called it successful calling when the passenger node finds the empty taxi node. In Figure 3, we could see that when the maximum speed of taxi is equal 60 km/h, it has the highest

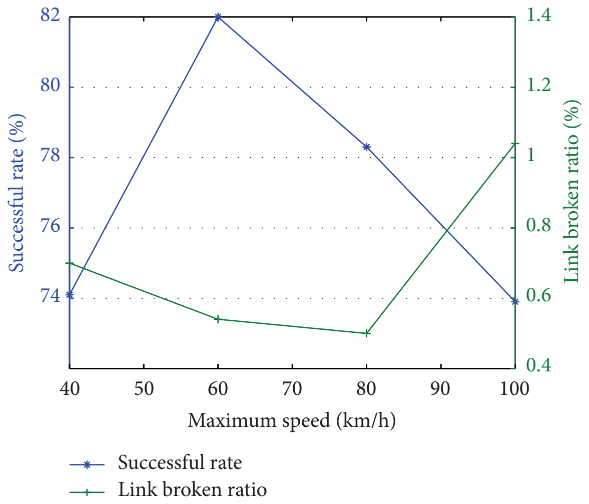


FIGURE 3: The relation of the maximum speed of taxi nodes and the ratio of passenger node contacting empty taxi nodes successfully.

successful ratio which enables the passenger node to find the empty taxi nodes. This is because when the node's moving speed is low, the node can contact with fewer other nodes. And when the node's moving speed is high, the network topology changes fast, which causes fewer available routing link and thus causes lower successful calling.

Figure 4 shows the impact of node's moving speed on the amount of switching packets of entire network.

As shown in Figure 4, the amount of switching packets grows up as the vehicle movement speed increases. This is because when the speed of node movement increases, it causes more correcting message, resulting in the increase of network data traffic. In addition, in one simulation, the routing overhead is higher at the beginning and gradually declines over time. This is because when the simulation starts, all taxi nodes are active at the same time and all taxi nodes broadcast at the same time, which causes more channel collision, leading to an increasing number of retransmissions. As time progresses, due to the correcting message, the taxi routing maintenance cycles gradually stagger and the entire network routing overhead declines.

Then we adjust the density of taxi nodes in the region. The average density of taxi nodes is set at 1–6 per square kilometer and the vehicle maximum speed is set at 60 km/h and then the results are shown in Figure 5.

As shown in Figure 5, when the taxi nodes are scarce, the rate of the success calling drastically reduces. While increasing the number of taxi nodes, the rate of success calling improves accordingly. This is because when the number of taxi nodes is too small, there is no taxi node around passenger node so that the passenger node cannot contact with any node, resulting in more calling failure. When the number of taxi nodes increase and, the passenger node sends an appointment request, the number of alternative routing paths increases, so the node can choose a more stable routing path to delivery packet; thereby the rate of success calling will increase.

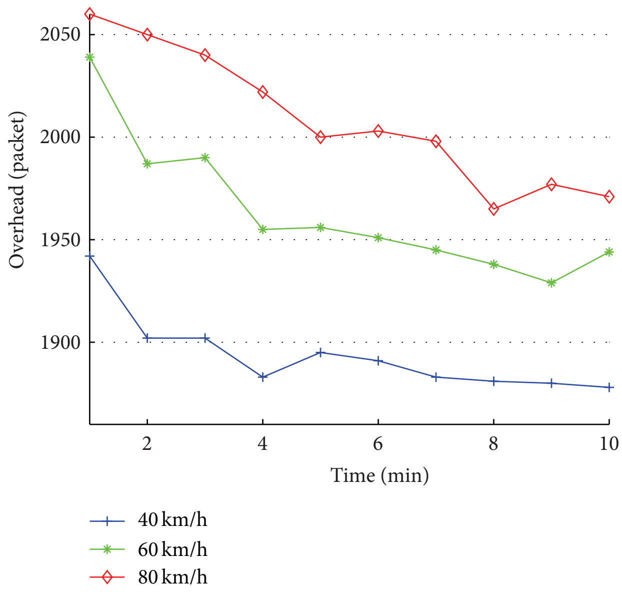


FIGURE 4: The impact of node's moving speed on the amount of switching packets of entire network.

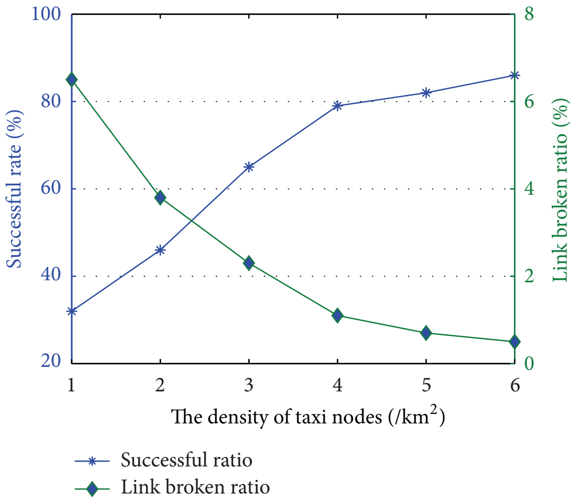


FIGURE 5: The impact of nodes density on successful ratio.

TABLE 2: The NS2 parameters.

Parameters	Values
Loss model	Propagation/TwoRayGround
Physical model	Phy/WirelessPhyExt
Antenna	Antenna/OmniAntenna
Transmission rate	2 Mbps
Network protocol	IEEE 802.11

At last, we control node velocity between 0 and 20 m/s; interval of T_b 10 s, transmission range is 500 m, and the density is 4 per square kilometers. Compared to AODV (show as Table 4), the results are shown in Table 2. We can see that the proposed routing protocol performs better than

Table 3: The impact of nodes density and transmission range on successful calling ratio.

Transmission range	Density of taxi			
	4		6	
	The probability to find T_D	The probability to receive from T_D	The probability to find T_D	The probability to receive from T_D
200 m	14.8%	78.5%	49.8%	95.5%
300 m	44.3%	91.2%	78.4%	94.2%
500 m	61.1%	94.1%	87.3%	94.6%

Table 4: Compare with AODV.

	CBLIGR	AODV
Delivery rate/(%)	61.1	52.3
Control message/packet	1206.8	1815.0

AODV. More importantly, our protocol communication data is scattered throughout the network, while AODV is relatively concentrated. As a result, our protocol performs better in the channel collision. It is because AODV requires a great deal of broadcast to maintain the entire link routing information and has no efficient methods to deal with the disconnection of links.

5. Conclusion

This paper proposes a routing protocol for taxi-calling application. Firstly, it analyzes scenes of taxi-calling application and studies the differences between the common vehicle application scene and the taxi-calling application scene. Secondly, it proposes the routing information maintaining algorithm based on position prediction which performs better reliability. Thirdly, a self-network forming method for taxi-calling application is drawn up.

Our protocol queries the destination node by nodes' keeping the information of the hop count to destination and decreases the routing cost. Meanwhile, the validity of routes is guaranteed by predicting the future position of themselves and their neighbors.

This protocol has also been evaluated by a traffic simulator. The results indicate that our protocol can efficiently find the target nodes in the area. It has also been found that the number of broadcasts is affected by nodes' velocity. And the fact that our protocol generates 30% less messages than AODV has been proven here.

As to future work, our protocol can probably be extended to solve the low delivery rate problem when the density of vehicle is low.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

This work is supported in part by the Natural Science Foundation of China "Research on the snapshot data security storage technology for authorization of release," no. 61100057.

References

[1] C. Rezende, R. W. Pazzi, and A. Boukerche, "An efficient neighborhoood prediction protocol to estimate link availability in VANETs," in *Proceedings of the 7th ACM International Symposium on Mobility Management and Wireless Access (MobiWAC '09)*, pp. 83–90, Tenerife, Spain, October 2009.

[2] T. Zahn, G. O'Shea, and A. Rowstron, "Feasibility of content dissemination between devices in moving vehicles," in *Proceedings of the 5th International Conference on Emerging Networking Experiments and Technologies (CoNEXT '09)*, pp. 97–108, Rome, Italy, December 2009.

[3] N. Wisitpongphan, F. Bai, P. Mudalige, and O. K. Tonguz, "On the routing problem in disconnected vehicular ad Hoc networks," in *Proceedings of the 26th IEEE International Conference on Computer Communications (IEEE INFOCOM '07)*, pp. 2291–2295, Anchorage, Alaska, USA, May 2007.

[4] A. Boukerche, H. A. B. F. Oliveira, E. F. Nakamura, and A. A. F. Loureiro, "Vehicular Ad Hoc networks: a new challenge for localization-based systems," *Computer Communications*, vol. 31, no. 12, pp. 2838–2849, 2008.

[5] C. E. Perkins and P. Bhagwat, "Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers," in *Proceedings of the Conference on Communications Architectures, Protocols and Applications (SIGCOMM '94)*, pp. 234–244, 1994.

[6] T. Spyropoulos, T. Turletti, and K. Obraczka, "Routing in delay-tolerant networks comprising heterogeneous node populations," *IEEE Transactions on Mobile Computing*, vol. 8, no. 8, pp. 1032–1047, 2009.

[7] C. Lochert, M. Mauve, H. Füßler, and H. Hartenstein, "Geographic routing in city scenarios," *Mobile Computing and Communication Review*, vol. 9, no. 1, pp. 69–72, 2005.

[8] M. Jerbi and S. M. Senouci, "Towards efficient routing in vehicular Ad Hoc networks," in *Proceedings of the 3rd IEEE International Workshop on Mobile Computing and Networking*, 2006.

[9] H. Menouar, M. Lenardi, and F. Filali, "An intelligent movement-based routing for VANETs," in *Proceedings of the ITS World Congress*, London, UK, October 2006.

[10] K. Shafiee and V. C. M. Leung, *Connectivity-Aware Minimum-Delay Geographic Routing with Vehicle Tracking in VANETs*, Elsevier B.V., 2010.

[11] J. Haerri, F. Filali, and C. Bonnet, "Performance comparison of AODV and OLSR in VANETs urban environments under realistic mobility patterns," in *Proceedings of the Med-Hoc-Net 2006, 5th Annual Mediterranean Ad Hoc Networking Workshop*, Lipari, Italy, June 2006.

[12] C. E. Perkins and E. M. Royer, "Ad-hoc on-demand distance vector routing," in *Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications (WMCSA '99)*, pp. 90–100, New Orleans, La, USA, February 1999.

[13] T. W. Chen and M. Gerla, "Global state routing: a new routing scheme for Ad-hoc wireless networks," in *Proceedings of the IEEE International Conference on Communications (ICC '98)*, vol. 1, pp. 171–175, Atlanta, Ga, USA, June 1998.

[14] B. Karp and H. T. Kung, "GPSR: greedy perimeter stateless routing for wireless networks," in *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking (MOBICOM '00)*, pp. 243–254, Boston, Mass, USA, August 2000.

[15] D. B. Johnson, D. A. Maltz, and J. Broch, "DSR: the dynamic source routing protocol for multi-hop wireless Ad hoc networks," in *Ad Hoc Networking*, C. E. Perkins, Ed., chapter 5, pp. 139–172, Addison-Wesley, New York, NY, USA, 2001.

[16] D. B. Johnson, D. A. Maltz, and J. Broch, "DSR: the dynamic source routing protocol for multi-Hop wireless ad hoc networks," in *Ad Hoc Networking*, C. E. Perkins, Ed., chapter 5, pp. 139–172, Addison-Wesley, New York, NY, USA, 2001.

[17] Z. Wei and S. Weibin, "Research of GPSR routing protocol for wireless sensor networks," *Electronic Measurement Technology*, vol. 33, no. 9, 2010.

[18] G. Wei, Y. Chen, and X. Zhu, "A vanet-oriented routing protocol for intelligent taxi-call systems," in *Proceedings of the IEEE 14th International Conference on Communication Technology (ICCT '12)*, pp. 41–45, Chengdu, China, November 2012.

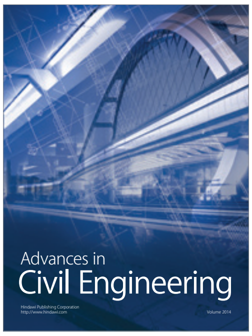
Journal of
Engineering

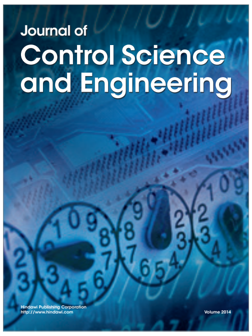
The Scientific World Journal

International Journal of
Rotating Machinery

Journal of
Sensors

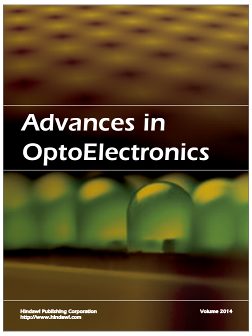
International Journal of
Distributed Sensor Networks


Advances in
Civil Engineering


Journal of
Control Science and Engineering

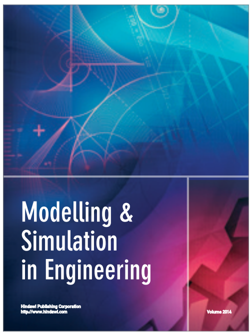
Journal of
Robotics

Journal of
Electrical and Computer Engineering

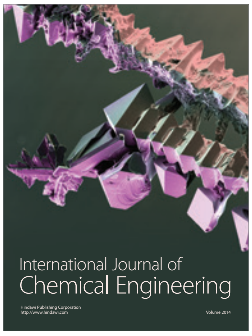

Advances in
OptoElectronics

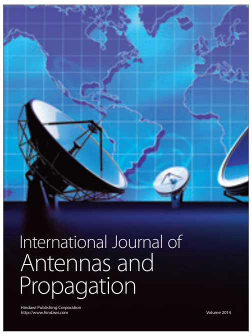
VLSI Design

International Journal of
Navigation and Observation

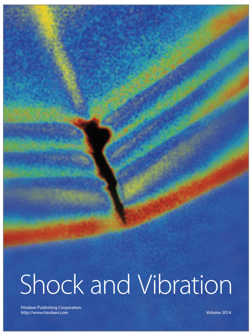

Modelling & Simulation in Engineering

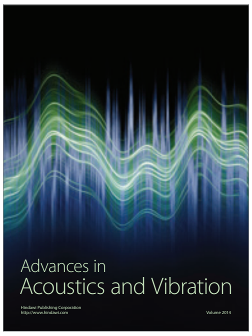
International Journal of
Aerospace Engineering


International Journal of
Chemical Engineering


International Journal of
Antennas and Propagation

Active and Passive Electronic Components


Shock and Vibration


Advances in
Acoustics and Vibration