Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 809742, 7 pages
http://dx.doi.org/10.1155/2013/809742

Research Article

A Study on the Tracking Problem in Vehicular Ad Hoc Networks

Xing Zhang, Bang Liu, and Jinchuan Tang

Department of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China

Correspondence should be addressed to Bang Liu; bangliuox1989@gmail.com

Received 3 December 2012; Accepted 12 January 2013

Academic Editor: Ming Liu

Vehicles have the characteristic of high mobility which makes vehicular ad hoc networks (VANETs) different from other mobile ad hoc networks (MANETs), it is more difficult to establish an end-to-end route in VANETs, and the source and destination nodes keep moving fast from their original locations. To guarantee a data packet will finally be received by the destination node in VANETs, and the location of the destination node must be tracked constantly. Advanced Greedy Forwarding (AGF (Naumov et al. (2006))) is a good way to estimate the location of the destination node by taking advantage of the velocity vectors; however, a vehicle changes its velocity constantly, it would be very inaccurate to estimate the location of a vehicle via its initial velocity. In this paper, we study the tracking problem in VANETs and propose two solutions: area-based tracking (ABT) and parked vehicle-assisted tracking (PVAT). ABT works well when the delays in data transmission from source to destination are small, and PVAT works as a supplement of ABT and deals with the situation of which the delays are large. PVAT takes advantage of vehicular sensors to detect whether a vehicle is parking. Simulations show better results compared to AGF.

1. Introduction

Routing is a great challenge in VANETs because of the high mobility of vehicles. Unlike routing in some static networks or other MANETs that the relay nodes in the routing path are static or moving slowly, the relay nodes in VANETs change their locations constantly and rapidly, and it is hard to find a relatively stable routing path in VANETs. However, the worst thing of routing in VANETs is not the location change of relay nodes, but the location change of the destination node. It is acceptable to find another relay node as long as it can deliver the data packet to the destination node finally, but it is not possible to change the destination node. Therefore, how to track the location of moving destination node and make sure the data packet can be finally delivered to it is an important topic in routing research of VANETs.

Traditional routing algorithm in ad hoc network is based on node topology, each node will maintain a routing table on a global view [1], and any changes on the neighbor node will cause updates on all routing tables. Because the routing table is global, the tracking of the destination node is not necessary, and data packets from the source node will arrive at the destination node in a very low latency. However, the overhead of maintaining the routing table is very big; when the network scale increases, the overhead will become unacceptable. Although some improvements have been made, like some reactive routing protocols [2, 3], the overhead is still proportional to the network scale.

Geographic routing is a routing that each node knows its own and neighbor node geographic position by position determining services like GPS [4]. It does not maintain any routing table or exchange any link state information with neighbor nodes [4]. Taking Greedy Perimeter Stateless Routing (GPSR), for example, in GPSR [5], the source node knows the location of the destination node, and the source node checks its neighbor table whether the destination node is in it; if the destination node is one of the source node's neighbor, the source node will deliver the data packets to the destination node, otherwise the source node will choose a neighbor which is closest to the destination node geographically. Same process happens on the relay nodes repeatedly until the destination node received the data packets, or delivery time expired. GPSR only needs partial information from neighbor nodes, therefore it has a very low overhead. However, GPSR has the tracking problem. The source node obtains the location of the destination node in the start of data transmission; when the data packets arrive at the location, the destination node has moved away already from the location in a big possibility. To address this problem, AGF is proposed in [6].

AGF adds a velocity vector to the context of the HELLO beacon packet which is used to inform the existence of a node to its neighbors. The velocity vector contains the moving direction and the speed of a node. The source node knows the velocity vector and the location of the destination node in advance. When a relay node receives a data packet, it will add the elapse time of the data packet sending from the source node to here to the processing time of itself; it will multiply the time result by the speed in the velocity vector of the destination node to get an estimation of the location of the destination node with the help of the moving direction also. It will check whether the estimated location is in the neighbor range, and the destination node is in the neighbor table; if condition satisfied, it will deliver the data packet to the destination, otherwise it will choose a neighbor which is closest to the estimation location as a next hop. AGF considers the movement of each node, therefore it obtains much higher data delivery rate compared to GPSR. However, velocity varies constantly, using a static value of velocity to estimate the location of the destination node is very inaccurate. As Figure 1 shows, AGF performs well in an origin case that no speed and direction varied. When speed begins to jitter, the performance starts to degrade. Because AGF has a searching process when the destination node is supposed to be in a neighbor table but it is not, AGF can handle some speed-variation cases when the whole packet delivery time is short, and the destination node is not far from its estimated location. However, when there is a delay in the whole packet delivery process and the destination node drives far away from its estimated location, the packet delivery will fail. Direction change is much worse than speed change to the performance of AGF, because when a vehicle makes turns, it is hard to track it according to its original direction. AGF uses broadcast to search for the destination node; however, wireless signal will be blocked by the obstacles by the road side as Figure 2 shows, like trees and buildings; if a vehicle wants to send a packet to a road segment on the orthogonal direction, unless it is in the center of the crossroad, the packet will not be able to penetrate the obstacles.

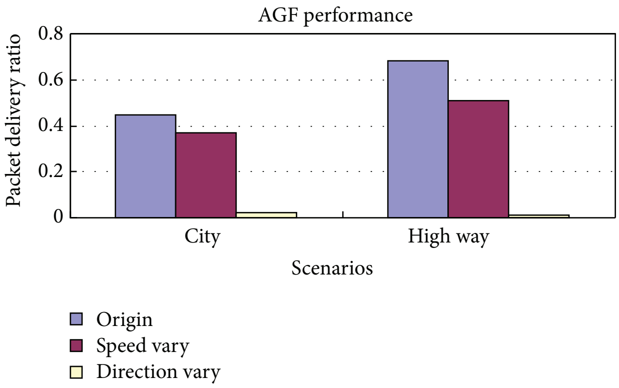

Figure 1: We use a Gaussian distributed random variable v_x with v_i mean and standard deviation σ. In city scenario, v_i equals to 40 kilometers per hour (km/h), and σ is 30 km/h; in high way scenario, v_i equals to 80 km/h, and σ is 10 km/h. The density is settled to a medium one. Detailed simulation setup is in Section 3.

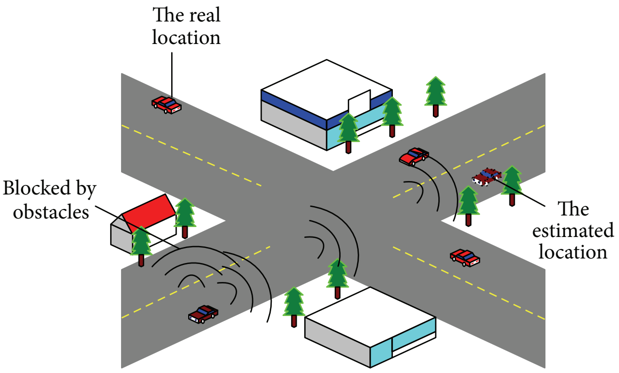

Figure 2: A vehicle is supposed to appear at the estimated location, and it turns left at the crossroad in fact. Another vehicle initiates a broadcasting to search for the vehicle, but the wireless signal is blocked by the obstacles by the roadside.

To remedy the disadvantages of AGF, we propose two tracking methods: ABT and PVAT. The basic idea of ABT is that instead of maintaining velocity vectors, ABT uses statistical results. The source node will choose an optimum route to the destination node based on traffic density according to VADD [7], see in Figure 3, the time of which the data packet travels on the selected route will be estimated. Therefore, we can calculate a distance of which how far the destination node moves away during the packet delivery by multiplying the time with the average speed of vehicles on the selected route. Instead of estimating the location of the destination node, ABT estimates an area of which the destination node may appear by considering all directions the destination node may head to. As Figure 4 shows, the area is rectangular and centers at the original location of the destination node. When a data packet arrives at the rectangular area, it will propagate inside the area until it meets the destination node, or delivery time expired. ABT does not need the information of moving speed and direction, therefore the performance will not suffer from speed and direction variation. However, ABT is at the expense of increasing packet redundancy, if the distance traveled by the destination node is long, the area will be large and the redundancy will be large, and therefore we propose another tracking method PVAT to deal with this problem. Vehicles' movement is restricted by road topology, and direction change of vehicle mostly happens at the intersection except for u turn. If every time the destination node passes by an intersection, the intersection will record the moving direction, and the destination node can easily be tracked. However, it is not easy to keep the information to a certain place in VANETs, because vehicles are highly mobile. [8] uses moving vehicles to cache the information at a certain location; once a vehicle with the information is moving out of the location area, it will pass the information to another vehicle which is in the location area. Vehicle density is changing constantly; when a vehicle leaves its location, it

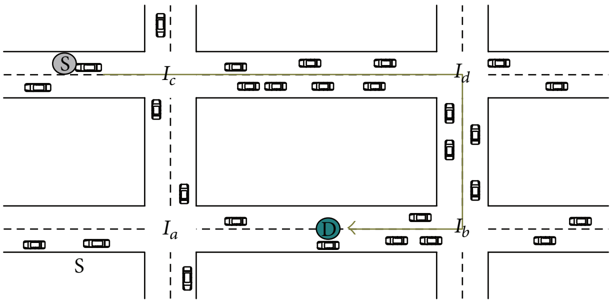

Figure 3: Traffic density-based route selecting. The shortest path from S to D is $I_c \longrightarrow I_a \longrightarrow I_b$, but the density on the path is low, so $I_c \longrightarrow I_d \longrightarrow I_b \longrightarrow I_a$ is chose.

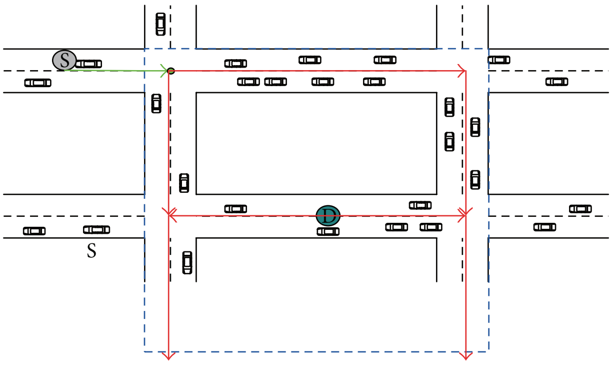

Figure 4: Area-base routing. The data packet will be delivered along the route selected as Figure 3 (green line); once it enters the area (blue rectangle), the data packet will spread over the whole area (red lines).

is not certain that it can find a substitution; also it is a big communication cost that the information keeps transmitting and receiving. Parked vehicles are wide spread in the city, and they can provide a good cache [9]. Parked vehicles are static and wide spread, therefore they can guarantee a more stable and costless cache than [8].

The rest of the paper is organized as follows: detailed descriptions of ABT and PVAT are in Section 2; performance evaluation is in Section 3; related works and conclusion is in Section 4.

2. Area-Based Tracking and Parked Vehicle-Assisted Tracking

ABT and PVAT aim to solve the tracking problem in VANETs, therefore they are more like patches to the existing routing algorithm. Before discussing about the detail of ABT and PVAT, we will describe the routing algorithm we use.

2.1. Traffic Density-Based Routing. Traditional geography-based routing like GPSR finds a neighbor node which is closest to the destination node, as a next hop. In theory, if the traffic density is satisfied, a data packet will finally reach the destination in GPSR. However, it is not optimum; because vehicles travel along road, a neighbor node which is closest to the destination in Euclidean distance may not be the closest node in the road topology. VADD is a routing algorithm based on road topology, and it achieves better performance than traditional geographic routing, therefore we use VADD as the main reference in routing.

However, VADD is a dynamic routing which decides the next hop of data delivery according to real situation, ABT needs to estimate the time for data delivery before real data sending, and therefore before real data routing using VADD, we select an optimum route based on statistical traffic density and calculate the elapse time according to (1).

Suppose Φ is the set of all connected path from the source node to the destination node; $W(x)$ gives the sum of each subpath's weight of path x; $N(x)$ is the number of subpath of x; $t(x)$ is the time elapse of data delivery on path x; l_i is the Euclidean distance of subpath i; ρ_i is the traffic density of subpath i; v_i is the velocity of vehicles on subpath i; R is the communication range of wireless signal, then

$$
\begin{aligned}
x &= \arg\max W(i), \quad i \in \Phi, \\
t_j(x) &= \begin{cases} \alpha \cdot l_j & \text{if } \dfrac{1}{\rho_j} \le R, \\[2ex] \dfrac{l_j}{v_j} - \beta \cdot \rho_j & \text{if } \dfrac{1}{\rho_j} > R, \end{cases} \\
t(x) &= \sum_{k=1}^{N(x)} t_k(x).
\end{aligned}
\tag{1}
$$

The equation chooses an optimum path x which has the maximum sum of traffic density, because the higher the traffic density, the better for data relay. If the average distance between vehicles is smaller than R, wireless transmission is used to forward the packet; otherwise, vehicles are used to carry the data; even in this case, it is still possible to occasionally have wireless transmissions; hence β is used as a correction factor [7]. The whole time elapse of data delivery is the sum of the time of each sub-path.

2.2. Area-Based Tracking. The basic idea of ABT is that when a data packet enters into the area of ABT, the data packet will be delivered to all roads in the area; as long as the destination node is in the area, it will finally receive the data packet. ABT needs to settle two challenges: how to set the area and how to propagate a data packet in the area.

2.2.1. Area Setting. In Section 2.1, an estimated time elapse of data delivery is given as $t(x)$, we can get an average vehicle speed in urban traffic $\overline{v}$, and therefore an estimated distance that the destination node may move away can be calculated as $d = t(x) \cdot \overline{v}$. The area can simply be a rectangle that it centers at the original location of the destination node and uses $2\lambda d$ as the length and width of the rectangle, and λ is an elastic coefficient which can be adjusted according to real situation.

However, the original rectangle may have to adjust to fit the road topology. As Figure 5 shows, sometimes the area

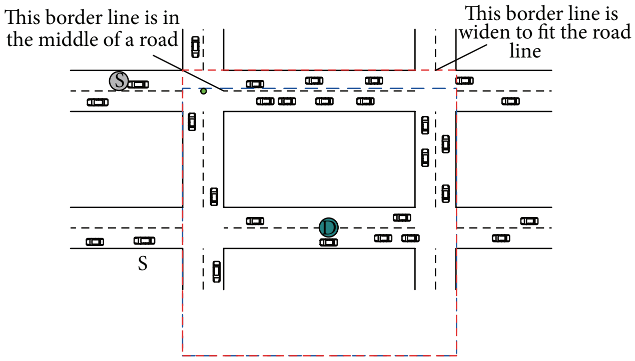
This border line is in the middle of a road
This border line is widen to fit the road line
S
D
S

Figure 5: Border line is in the middle of a road.

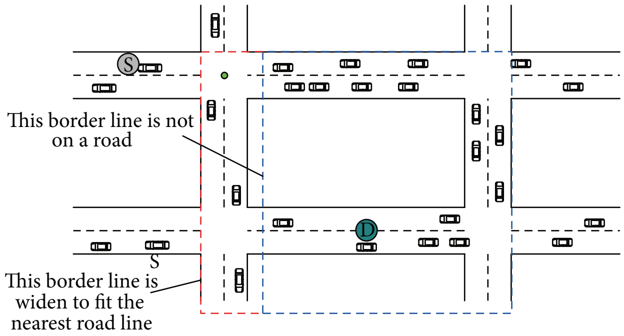
S
This border line is not on a road
D
S
This border line is widen to fit the nearest road line

Figure 6: Border line is not on a road.

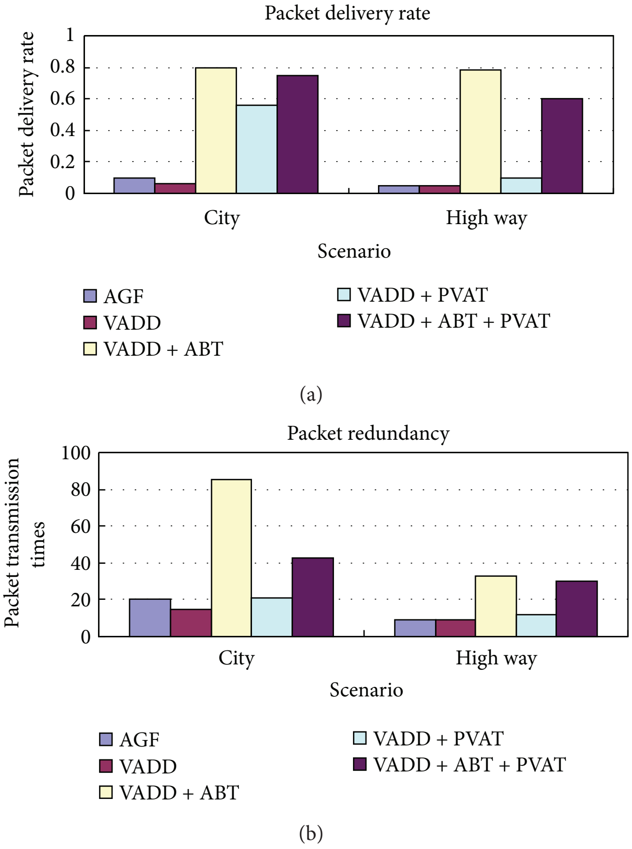

Figure 7: Simulation results.

border line is in the middle of a road. In this case, the border line will be widen to fit the road line. As Figure 6 shows, sometimes the area border line is not on any road. In this case, the border line will be widened or narrowed down to its nearest road line.

2.2.2. Data Packet Propagation. Once a data packet enters into the area, ABT will spread the packet all over the area to guarantee that the destination node will receive the packet. Epidemic routing [10] is suitable to achieve the goal. To reduce unnecessary packet redundancy, ABT makes sure that if a node already has the packet, it will not receive the packet again; if a node is not in the area, it will not receive the packet; if a node already delivered the packet to the destination node, it will not send the packet to another node. DBT algorithm is given in Algorithms 1 and 2.

If a node needs to deliver a packet, it will traverse its neighbor table; if a neighbor is not in the area scale, it will not deliver the packet; if a neighbor is in the area scale, it will send a request to the neighbor; the neighbor will check whether the packet has already been received, if true, reply ALLOW, else reply DISALLOW; if the neighbor allow to send packet, the sender will send the packet; if the neighbor is the destination node, the sending process will be end. After looptime ends, if the packet still has not been delivered to the destination node, and the sending process will be end.

2.3. Parked Vehicle-Assisted Tracking. Parked vehicles by the roadside are very common in city environment, [9] mentioned to use these parked vehicles as data delivery relay nodes. Parked vehicles are static nodes which can provide stable communication and data cache. If a data packet needs to be cached to a location, using moving vehicles will cause the data packet been delivered over and over again when the data-carry vehicle moves out of the location area. Parked vehicles do not leave an area as frequent as the moving vehicles do, therefore using parked vehicles will greatly reduce the overhead of caching data. Moreover, parked vehicles are wide spread in the city, especially in the night that traffic density is very low, the density of parked vehicles are very high, and they can provide good data delivery which cannot be achieved by using moving vehicles.

To detect whether a vehicle is parking or not, PVAT takes advantage of vehicular sensors. OBD scanner can read revolutions per minute (RPM) of vehicle engine which can be used to indicate the parking state of a vehicle.

PVAT records the direction information of the destination node at the intersections when the destination node passes by, it is similar to [8], and the difference is that PVAT uses parked vehicles which can provide low overhead data cache and high reliability of vehicle tracking, because moving vehicles suffer from nonuniform traffic density and traffic holes [11] which will cause data caching fail. The PVAT algorithms are given in Algorithms 3 and 4.

In PVAT, when a node receives a packet, it will firstly check whether the destination node is in its communication

```
Require:
    data packet, area scale, destination node ID, period, looptime.
Ensure:
    true, false.
    repeat
        for each node in the neighbor table
            if node is not in area scale
                continue;
            end if
            send request for data packet delivery;
            if wait for response time out
                continue;
            end if
            if response == ALLOW
                send the data packet;
                if node ID == destination node ID
                    return true;
                end if
            end if
        end for
        wait for period time;
    until looptime-- == 0;
    return false;
```

Algorithm 1: Sender of area-based tracking.

```
Require:
Ensure:
    while 1
        if received a request
            if the data packet according to the request
            has already been received
                send DISALLOW response;
                continue;
            end if
            send ALLOW response;
        wait to receive the data packet
        end if
    end while
```

Algorithm 2: Receiver of area-based tracking.

range and send data packet to it, otherwise it will check if it is in an intersection, because that is where the destination node leaves its marks. If marks are found, the data packet will be delivered to the nodes holding the marks and continue data delivery using VADD to the other end of the road segment until no marks are found, or the data packet is delivered to the destination node.

2.4. VADD Plus Tracking Solutions. In this Subsection, we give a whole routing solution combined VADD, ABT, and PVAT, see Algorithm 5. Each packet has a *flag* bit which indicates whether PVAT should be activated. ABT has more packet redundancy than PVAT, which increases communication overhead. It seems that PVAT is better than ABT, but neither parked vehicle nor moving vehicle may fail to cache the marks of the destination node because there are no such vehicles there for caching sometimes. Although ABT uses more packet redundancies, it has much better data delivery rate than PVAT, and therefore we use ABT as main tracking method and PVAT as a supplement way.

3. Performance Evaluation

In this paper, we use ns-2 simulator for simulations. We test packet delivery rate and packet transmission times of AGF, VADD, VADD plus ABT, VADD plus PVAT, and VADD plus ABT and PVAT. We write the AGF and VADD codes according to [6, 7].

```
Require:
Ensure:
  while 1
    mapping its location from GPS to the road segment in the map;
    compare the road segment with the previous road segment;
    if it is in a different road segment
      send destination ID to parked vehicles;
      if no response from parked vehicles
        send destination ID to moving neighbors;
        if no response from moving neighbors
          return;
        end if
      end if
    end if
    wait for a period;
  end while
```

Algorithm 3: Marks leaving of the destination node.

```
Require:
  data packet, location
Ensure:
  newlocation
  if the destination node is in the communication range
    send the data packet to the destination node;
    return 0;
  end if
  if in an intersection
    request neighbor vehicles including parked and
    moving vehicles which have the marks of the
    destination node to reply their locations;
    map the locations to the road segment in the map;
    //the replied locations should be one end of the road segment.
    send the data packet to the replied nodes;
    return the location of the other end of the road segment;
  end if
  send data packet to the node of next hop according to VADD;
  return location
```

Algorithm 4: Parked vehicle-assisted tracking.

We use a similar simulation setup of AGF. The radio propagation model that is using $[P_r(d)/P_r(d_0)]_{db} = -10\beta \log(d/d_0) + X_{db}$, X_{db} is a Gaussian distributed random variable with zero mean and standard deviation σ_{db}; β is the path loss exponent, and $P_r(d_0)$ is the reference received power for a close-in distance d_0 [6]. σ_{db} is set to 8, and β is set to 2 in city scenario and 4 in high scenario. The communication range is set to 550 meters, and 802.11b protocol is used for communication. Sources start generating data packets within the first 50 s of the simulation time and stop generating data packets 50 s before the simulation ends to avoid data packet drops due to the end of the simulation; all source/sink nodes stay inside the simulated area for the duration of their communications [6]. Speed variation and direction variation are same as Figure 1.

We can see from Figure 7 that VADD plus ABT plus PVAT achieve high packet delivery ratio, meanwhile it costs much small packet redundancy comparing to VADD plus ABT.

4. Conclusions

In this paper, we propose two solutions of the tracking problems existing in VANETs routing. ABT that has good packet delivery rate yet costs high packet redundancy, and PVAT costs less redundancy, but the packet delivery rate is not as good as ABT. We combine VADD, ABT, and PVAT to achieve both high packet delivery rate and low packet redundancy.

```
Require:
    data packet, destination location, destination node ID, location
Ensure:
    generate an area A according to (1);
    if size of A is smaller than B
        if destination location not belongs to A
        use VADD to find the next hop;
            return;
        end if
        use Algorithm 1 for data delivery; //call ABT(data packet, A, destination node ID)
        return;
    else
        if flag == PVAT
            use Algorithm 4 for data delivery; //call PVAT(data packet, location)
            return;
        else
            if location is in the communication range of destination location
                set flag to PVAT;
                use Algorithm 4 for data delivery; //call PVAT(data packet, location)
                return;
            else
                use VADD for data delivery;
                return;
            end if
        end if
    end if
```

Algorithm 5: VADD plus ABT plus PVAT.

Acknowledgment

This work is supported by National Science Foundation under Grant nos. 60903158, 61003229, 61170256, 61103226, and 61103227 and the Fundamental Research Funds for the Central Universities under Grant nos. ZYGX2010J074, ZYGX2011J073, ZYGX2011J060, and ZYGX2011J102.

References

[1] G. Pei, M. Gerla, and T. W. Chen, "Fisheye state routing: a routing scheme for ad hoc wireless networks," in *Proceedings of the IEEE International Conference on Communications*, pp. 70–74, June 2000.

[2] 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, February 1999.

[3] D. Johnson and D. Maltz, "Dynamic source routing in Ad Hoc wireless networks," in *Mobile Computing*, pp. 153–181, Springer, New York, NY, USA, 1996.

[4] S. Jayachandran, J. Jothi, and S. Krishnan, "A case study on various routing strategies of VANETs," in *Global Trends in Computing and Communication Systems*, pp. 353–362, Springer, Berlin, Germany, 2012.

[5] 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.

[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 (MOBIHOC '06)*, pp. 108–119, May 2006.

[7] J. Zhao and G. Cao, "VADD: vehicle-assisted data delivery in vehicular Ad hoc networks," *IEEE Transactions on Vehicular Technology*, vol. 57, no. 3, pp. 1910–1922, 2008.

[8] K. Shafiee and V. C. M. Leung, "Connectivity-aware minimum-delay geographic routing with vehicle tracking in VANETs," *Ad Hoc Networks*, vol. 9, no. 2, pp. 131–141, 2011.

[9] N. Liu, M. Liu, W. Lou, G. Chen, and J. Cao, "PVA in VANETs: stopped cars are not silent," in *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM '11)*, pp. 431–435, April 2011.

[10] A. Vahdat and D. Becker, "CS-200006," Tech. Rep., Duke University, 2000.

[11] C. Song, M. Liu, Y. Wen et al., "Towards the traffic hole problem in VANETs," in *Proceedings of the 9th ACM International Workshop on Vehicular Inter-Networking, Systems, and Applications, Low Wood Bay*, pp. 139–140, Lake District, UK, 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