

# TIGeR: a Traffic-Aware Intersection-based Geographical Routing Protocol for Urban VANETs

Rasool Tavakoli

Department of Electrical and Computer Engineering  
Isfahan University of Technology, Isfahan, Iran  
Email: r.tavakolinajafabadi@ec.iut.ac.ir

Majid Nabi

Department of Electrical Engineering  
Eindhoven University of Technology, the Netherlands  
Email: m.nabi@tue.nl

**Abstract**—In recent years, vehicular ad-hoc networks (VANETs) have received increasing interests because of their promising solutions in many urban applications. VANETs are distinguished from other kinds of ad-hoc networks, such as Mobile Ad-hoc Networks (MANETs), by high mobility and reduced communications. The topology of the network is highly time-varying due to high mobility of wireless nodes, and also joining and leaving nodes. This makes it challenging to find and maintain efficient and reliable data forwarding routes between different nodes in the network. On the other hand, equipment such as positioning systems usually exist in current vehicles which can be used to improve the efficiency of packet dissemination. This paper proposes TIGeR, a new Traffic-aware Intersection-based Geographical Routing protocol. Nodes that are at intersections make routing decision to forward packets based on both the local vehicular traffic information of different roads and the road's angle with respect to the direct vector toward the destination. Local vehicular traffic information is considered as a sign of network connectivity in a road. Simulation results show that TIGeR improves the packet delivery ratio, especially in sparse networks.

## I. INTRODUCTION

Vehicular ad-hoc network (VANET) is an emerging technology that exploits the wireless communication capability of modern vehicles to establish an infrastructure-free network. The main idea is to realize an Intelligent Transportation System (ITS) by providing efficient vehicle-to-vehicle (V2V) communication or road-to-vehicle communication (RVC). VANET can be used as a promising solution for a variety of applications such as traffic monitoring and control, disaster management, and urban supervision.

Vehicular ad-hoc networks are considered as a particular class of mobile ad-hoc networks (MANETs) built on moving vehicles. This kind of networks is characterized by high node mobility in limited degree of freedom and unreliable wireless channel conditions. On the other hand, nodes in these networks have sufficient energy resources and processing power to perform more complicated networking protocols. Such particular features often make standards and common MANET protocols sub-optimal for VANETs. Because of high vehicle mobility, the network topology and nodes' connectivity are continuously changing over time. The routing protocol is responsible to handle such topology changes while routing data packets. There are certain Quality-of-Service (QoS) requirements in every VANET application that should be satisfied by the routing algorithm. End-to-end latency and packet delivery ratio (PDR) are common QoS requirement.

VANET routing protocols are classified into *topology-based* and *position-based* classes. In topology-based approaches

(e.g., [1],[2]), the next forwarding node is selected based on the network topology information. Location-based or geographical routing protocols (e.g., [3],[4],[5]) are based on geographical information of the nodes. By knowing the location of the destination and the nodes in the network, forwarding nodes can be selected to route the packet from the source node to the desired destination(s). Application-level QoS demands, and the features of the hardware platform (e.g., communication and computation capabilities and peripheral equipment) should be taken into consideration for designing an appropriate routing technique for such multi-hop data propagation.

In this paper, we focus on geographic routing in city environments. As it is common in VANETs, we take into account the use of Global Positioning System (GPS) devices so that every vehicle is aware of its own location and movement parameters such as velocity and direction. We propose TIGeR, a new Traffic-aware Intersection-based Geographical Routing protocol. It uses an angular-based mechanism taking the local traffic information at intersections into account to select reliable routes toward the destination.

The rest of the paper is organized as follows. Related work on geographical routing is reviewed in Section II. Section III presents the proposed routing protocol. The performance evaluation results are given in Section IV. Section V concludes.

## II. RELATED WORK

One of the first geographical routing protocols for mobile environments is GPSR [3]. It works in two modes. 1) *Greedy forwarding mode* in which packets are forwarded to the nodes that are geographically closer to the destination. If there is not any node closer to the destination, the greedy forwarding may come to a deadlock. In such condition, the forwarding node switches to the second mode. 2) *Perimeter mode* in which the node forwards the packet to one of its neighbors based on the right-hand rule. The forwarding node continues in perimeter mode until it finds a neighbor that is closer to the destination.

Some mechanisms use nodes at intersections and street topology to make better routing decisions. For instance, GPCR [4] exploits an intersection detection mechanism using information of one- and two-hop neighbors. This information is obtained by periodically exchanging messages between neighboring nodes which imposes a high communication overhead. Geographic source routing (GSR) [6] is an intersection-based protocol in which the sender uses Dijkstra shortest path algorithm to determine a sequence of intersections that a packet needs to traverse toward its destination by using the street map. This information is then included in the packet header

to be used by all forwarding nodes in the path. Vehicle assisted data delivery protocol (VADD) [7] also uses street map. This protocol uses a store-and-forward technique at intersections to select the roads with the lowest estimated delay to the destination. In static-node assisted data dissemination protocol (SADV) [8], some static nodes at intersections are used to improve data delivery. When there is no vehicle available to forward a packet, the nearest static node stores the packet until a proper delivery path is available. A multi-path routing mechanism is also exploited in SADV to reduce data delivery delay at the cost of higher communication overhead.

None of the mentioned protocols takes the vehicular traffic information into account for making routing decisions. Therefore, they may lead to routing loops. The Greedy Traffic-aware Routing protocol (GyTAR) [9] takes the roads vehicular traffic into account to select the most reliable route. When a packet reaches a node at an intersection, the node selects an adjacent intersection toward the destination with the highest vehicular traffic, hoping that it provides better wireless connectivity. This leads to performance improvement comparing to other routing approaches. However, it uses a decentralized cellular traffic density estimation method which imposes high network traffic overhead. Moreover, routing may fail when no digital map is available.

We propose a new intersection-based geographical routing protocol for multi-hop data routing in vehicular networks for city roads. At intersections, nodes use both local traffic information in different roads and angle of roads aiming to find most reliable and efficient route for the given packet. Unlike most of the geographical routing protocols, our protocol does not need a digital map to find intersections. Instead, it develops a method using which each node dynamically detects whether it is at an intersection. This makes the TIGeR protocol also usable for places for which a digital map is not available.

### III. TIGER PROTOCOL

We assume that each vehicle in the network is equipped with a GPS and so knows its position and velocity vector. We do not use digital maps in our protocol, because it may be unavailable for some areas. We consider any wireless intersection (points that a node can wirelessly reach nodes in other roads) as intersection and use it for making routing decisions.

In geographical routing protocols, the source node needs to know the current position of the destination to start routing. We use ideal location service for this purpose. Each node maintains a neighborhood table, recording some information of its neighbors such as their position and velocity vector. This information is obtained from periodic hello messages that are exchanged between vehicles. The core idea of TIGeR is to deliver data packets in an urban environment by selecting proper routes at intersections. Intersections include road intersections, the points where a vehicle can establish a wireless connection to vehicles outside its main road, and the points where a street changes its direction. These points have a direct relation with street obstacles. Instead of using a digital map, we develop a mechanism to detect if a node is at an intersection situation.

#### A. Dynamic intersection detection mechanism

Assume  $N = \{N_1, N_2, \dots, N_j, \dots\}$  is the set of all nodes in the network, size of which ( $|N|$ ) is changing over time, as

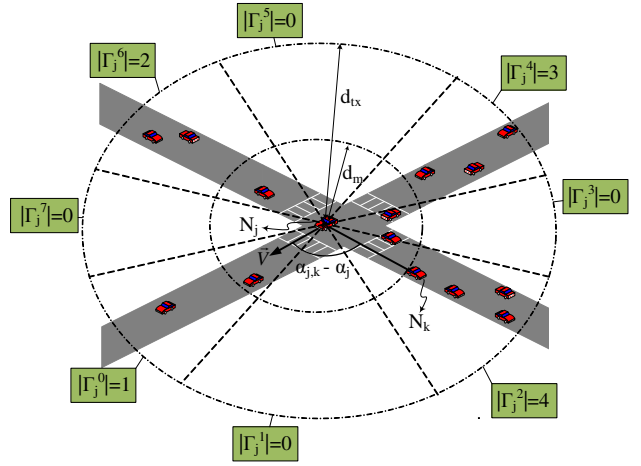


Fig. 1. Forming angular neighbor subsets in an illustrative situation.

some vehicles may join the traffic or leave the covered area of the network. Every time a node receives a hello message from one of its neighbors, its neighborhood table is updated. Let  $\Gamma_j \subset N$  be the current neighbor set of node  $N_j$ . This set includes all nodes that are within the transmission range ( $d_{tx}$ ) of node  $N_j$ . Each node  $N_j$  forms a number of non-overlapping sections around itself and investigates in which section the position of each neighbor lies (Fig. 1). Accordingly, a number of subsets are extracted, each containing all neighbors in different sections that are farther than a certain distance ( $d_m$ ) from  $N_j$ . The neighbor nodes near node  $N_j$  are excluded as their movement direction around the intersection is not predictable and they may change their direction very soon. So they cannot be reliable for intersection detection. For simplicity and w.l.o.g., we assume an angular resolution of eight sections for the rest of this paper. However, higher angular resolutions can also be applied which may lead to higher precision, but may impose more computation overhead. Eqn. 1 introduce the subset extraction paradigm.

$$\Gamma_j^i = \{N_k \in \Gamma_j : d_{j,k} > d_m \wedge (\frac{i\pi}{4} - \frac{\pi}{8}) \leq \alpha_{j,k} - \alpha_j < (\frac{i\pi}{4} + \frac{\pi}{8})\} \quad (1)$$

where  $d_{j,k}$  and  $\alpha_{j,k}$  denotes the Euclidean distance between node  $N_j$  and  $N_k$ , and the geographical angle between these nodes (without considering the movements' direction), respectively.  $\alpha_j$  stands for the angle of the movement vector of  $N_j$ . In the case that  $\alpha_{j,k} < \alpha_j$ ,  $2\pi$  is added to the value of  $\alpha_{j,k} - \alpha_j$  to keep it positive. Note that  $0 \leq i \leq 7$ , so that eight subsets are extracted.  $\Gamma_j^i \subseteq \Gamma_j$  is the  $i^{th}$  angular neighbor subset of node  $N_j$  and contains all neighbors of node  $N_j$  which are farther than  $d_m$  and their position lies in  $i^{th}$  angular section out of eight sections. These subsets are updated whenever a hello message is received. Fig. 1 shows the angular subsets in an illustrative situation. These subsets are used both for intersection detection and data forwarding route selection by nodes at intersections.

**Lemma1:** Node  $N_j$  is at a wireless intersection if there is at least one angular neighbor subset which has at least a certain number of nodes ( $k_{min}$ ) and it is not on the main road as node  $N_j$ .

$$N_j \text{ is intersection node} \Leftrightarrow \exists \Gamma_j^i : |\Gamma_j^i| > K_{min} \wedge i \neq 0, 4 \quad (2)$$

Note that subsets  $\Gamma_j^0$  and  $\Gamma_j^4$  contain vehicles in the same roads as  $N_j$  (in front of and behind node  $N_j$ ) and thus do not show an intersection. A node announces if it is an intersection node by including one bit in its hello messages. So all nodes are aware about the status of their neighbors.

Once a node receives a data packet, it first checks if the desired destination of the packet is one of its neighbors. In this case, it forwards the packet to the destination. Otherwise the next forwarding node should be selected. The selection mechanism in TIGeR depends on the fact that the node is intersection or non-intersection node. In the following, we explain the forwarding mechanism used for routing by intersection and non-intersection nodes.

### B. Forwarding mechanism for intersection nodes

When an intersection node receives a data packet and the destination of the packet is not among its direct neighbors, it tries to select the best candidate forwarding neighbor by using an angular-based vehicular traffic-aware algorithm. This algorithm uses angular neighbor subsets that were previously described. A relative score is computed for each angular neighbor subset  $\Gamma_j^i$  according to the geographical angle of  $N_j$  toward the destination, the mean angle toward the members of  $\Gamma_j^i$ , and the number of neighbors in each angular neighbor subset  $|\Gamma_j^i|$ . Then the subset that obtains the highest score is selected for packet forwarding. The goal of such scoring method is to select a road with the smallest angle w.r.t. the packet destination that has the highest number of nodes. A higher size of the subset is considered as a sign of a higher vehicular traffic and thus better wireless connectivity. Eqn. 3 gives the routing score for the angular neighbor subset  $\Gamma_j^i$ .

$$Score(\Gamma_j^i) = \frac{|\Gamma_j^i| \times (\pi - \phi_j^i)}{\phi_j^i + \epsilon}, \quad 0 \leq i \leq 7 \quad (3)$$

$|\Gamma_j^i|$  reflects the local vehicular traffic in a road (Fig. 1). We assume that the vehicular traffic of a street is uniformly distributed between two end intersections. Thus we can have an estimation of the overall traffic of a road by knowing its local traffic. Here we again emphasize that the angular neighbor subsets only include neighbors with a minimum distance of  $d_m$ . Therefore, all vehicles around the intersection area are ignored since they may not have a stable direction. Moreover, traffic lights make the part of the road around the intersections more crowded than other parts. So they are not taken into account for estimating overall vehicular traffic of the road.

The next influencing term in Eqn. 3 is  $\phi_j^i$  that represents the angular difference between the straight direction toward the destination and the mean direction toward the members of  $\Gamma_j^i$  (Fig. 2). This term is calculated using Eqn. 4.

$$\phi_j^i = \begin{cases} |\alpha_{j,dest} - \theta_j^i| & |\alpha_{j,dest} - \theta_j^i| < \pi \\ 2\pi - |\alpha_{j,dest} - \theta_j^i| & otherwise \end{cases} \quad (4)$$

$$\theta_j^i = \frac{1}{|\Gamma_j^i|} \sum_{N_k \in \Gamma_j^i} \alpha_{j,k} \quad (5)$$

where  $\theta_j^i$  is the *road angle* which is the average angle of all members of  $\Gamma_j^i$  w.r.t. node  $N_j$ . It shows the overall angle of

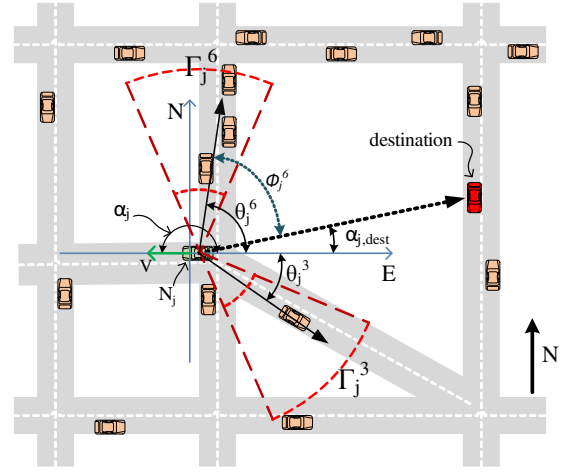


Fig. 2. The road angle calculation in the TIGeR protocol.

the road which corresponds to the angular neighbor subset  $\Gamma_j^i$ . Fig. 2 illustrates this angle difference in an example situation. Smaller value of  $\phi_j^i$  is preferred, because it shows that the angle of this angular neighbor subset w.r.t. the destination is smaller. Thus it is considered a better choice for routing the packet. According to this, whatever the term  $\phi_j^i$  at the denominator of Eqn. 3 has lower value, the score of  $\Gamma_j^i$  increases. If the value of the term  $\phi_j^i$  is close to  $\pi$ , it means that the direction of  $\Gamma_j^i$  is toward the opposite side of the location of the destination. Thus the term  $\pi - \phi_j^i$  in Eqn. 3 goes down to 0 and the score decreases. The term  $\epsilon$  is a small value (e.g.,  $\epsilon = \frac{\pi}{10}$ ) and is added to avoid zero dominator. Note that using this scoring method, the angular neighbor subset from which the data packet is received gets a very low score.

The path with highest score is selected for routing the packet. After selection of the proper angular subset, the farthest node in this subset that moves against the current intersection is selected and the data packet is forwarded to it. If there is not such a node with mentioned moving direction, farthest node on the road is selected. Before forwarding the packet, the angle of the selected road ( $\theta_j^i$ ) is added to the packet header. This field will be used for angular packet forwarding by the non-intersection nodes.

### C. Forwarding mechanism for non-intersection nodes

**Angular forwarding on roads:** like intersection nodes, when a non-intersection node receives a data packet, it checks if the destination is among its direct neighbors. If so, the packet is delivered to the destination. Otherwise, the node uses an angular forwarding strategy to help the packet to go forward along the road with the road angle  $\theta_j^i$ . The current non-intersection node  $N_j$  extracts the road angle from the packet (which is set by the last intersection node). Then it looks for its farthest neighbor along the direction of  $\theta_j^i$ . This node is expected to be the best candidate to be the next relaying node. This is done by any non-intersection node that receives the packet until the packet reaches the next intersection node or its destination.

The neighbor set of a node is updated by exchanging periodic hello messages. Such hello messages include information about the current position and movement vector of

the sender node of the message. Therefore, node  $N_j$  knows geographical position and movement vector (direction and velocity) of neighbor  $N_k \in \Gamma_j$  at the time of the last received hello message from  $N_k$ . When node  $N_j$  is looking for the next relaying node for a packet, its information might not be completely up-to-date, because the nodes may be moving. Some neighbors may have even gone outside the transmission range of node  $N_j$ . Because of that we need to exploit a position prediction method based on the last movement vector of each neighbor. We use the method presented in GyTAR [9] to predict the current position of the neighbors. By using this method, node  $N_j$  can also predict out of range nodes.

In the case that a non-intersection node is not able to find a node along the road angle attached to the packet, it starts performing a *carry and forward* process. The node carries the packet until the vehicle reaches an intersection or it finds a proper neighbor to deliver the packet. However, we consider a certain time limit for carrying a packet. If the node cannot find a proper node (either an intersection node or a neighbor along the road angle) within the specified time limit, the *route recovery* mechanism is started by the node.

**Route Recovery (RR):** A non-intersection node that has received a packet can be a local optimum for routing that packet. It means that this node does not have any neighbor along the road angle that can help in routing the packet. As it was mentioned, the node uses carry and forward, hoping to find a proper neighbor along the road angle or an intersection node in near future. However, it is possible that the node is in a dead-end street or a partitioned network situation, for instance. In such cases, carry and forward cannot resolve the problem. Because of that, the node is allowed to perform carry and forward only for a certain (short) period of time. After this period, the packet goes to route recovery mode. It means that the selected road by the last intersection node has failed and the packet should be sent back to the same intersection node to be routed through an alternative road. When a packet enters route recovery, a field is set in the packet header. Thus the other nodes are informed about it and try to route the packet at the opposite direction ( $2\pi - \theta_j^i$ ). This continues until the packet reaches the intersection. Now, at the intersection node, the routing decision is made ignoring the failed road (which the packet came back from).

#### IV. PERFORMANCE EVALUATION

To evaluate the performance of the TIGeR protocol, we implemented the protocol on top of the network simulator NS2.35 [10]. We ran various simulations using two versions of our protocol; a version fortified by the route recovery mechanism (TIGeR+RR), and another without route recovery (TIGeR-RR). This allows us to investigate the behavior of the route recovery mechanism. Moreover, to have a comparison with commonly used routing protocols, we also tried an implementation of GPSR with improved performance presented in [11] as well as AODV [2] which is a well-known topology-based routing protocol for ad hoc networks .

##### A. Simulation setup

We used a  $3500m \times 4000m$  part of Manhattan area which is extracted from TIGeR/Line database [12] to generate vehicular movement patterns for our simulations. This map contains

TABLE I  
SETTING USED FOR SIMULATIONS.

Parameter	Value
Simulation area	$3500m \times 4000m$
Number of vehicles	100 to 400
Number of CBR sources	20
Transmission range	500m
MAC protocol	IEEE 802.11p
MAC data rate	2 Mbps
Simulation time	400sec
Vehicle velocity	20-50 km per hour
CBR rate	1 to 5 packet per second
Data packet size	512 bytes
Propagation model	two ray ground

road information such as the number of street lanes and speed limits. We use Traffic and Network Simulation Environment TraNS v1.2 [13] for generating realistic vehicular movement patterns. TraNS uses SUMO [14] to generate random or user defined vehicular movement patterns based on the map. It defines some random vehicle flows between various points on the map and moves vehicles with random speeds within the given speed limit of the roads. We run our simulation with different vehicle densities varied from 100 to 400 vehicles. Twenty nodes are randomly selected as the source nodes generating CBR traffic with various rates of 1 to 5 packets per second. We used  $d_m = 50m$  and  $k_{min} = 2$ . Each packet remains in carry and forward mode only for five seconds at local optimums. Each simulation is repeated five times to have statistically more reliable results. Table I presents the value for some parameters of the simulation setup.

##### B. Results

We investigate various metrics to evaluate the performance of the protocol. Packet Delivery Ratio (PDR) is the percentage of generated data packets that reach their destinations. This metric reflects the reliability of the used routing protocol. Fig. 3 depicts the achieved PDR using various routing protocols for different number of nodes. This result shows that TIGeR provides better PDR in comparison with GPSR and AODV in both sparse and dense networks. This is because of taking into account the traffic density of the roads at intersections for selecting packet forwarding route. GPSR uses a greedy strategy which causes a higher chance of sticking at local optimums than TIGeR. Because of the high nodes mobility and unstable data routing paths in VANET scenarios, routing failure rate of AODV is also higher than TIGeR for all network densities. Note that in low density networks, network partitioning problem makes the probability of facing local optimums higher. The next observation of Fig. 3 is that the route recovery strategy of TIGeR+RR really helps improving the reliability of data dissemination.

Fig. 4 shows the obtained PDR for different packet generation rates. For higher rates, PDR value decreases in most cases. This is due to congestions (high packet traffic), considering physical and MAC layer limitations. Both versions of TIGeR provide considerably better PDR than GPSR and AODV for all packet rates. This is because TIGeR considers real-time vehicular traffic information as well for selecting routes with better connectivity. The result clearly shows that route recovery improves PDR and it means that local optimums occur during network operation which need to be resolved.

To better investigate the behavior of TIGeR, we measure the average number of hops that a packet traverses until it reaches its destination (path length). Fig. 5 shows the average

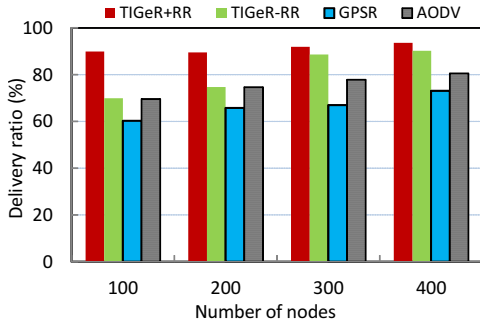


Fig. 3. Average PDR for different number of nodes (packet generation rate of 1 pkt/sec.)

path length of the delivered packets for different network densities. The figure shows that TIGeR has in general higher average path length than GPSR and AODV. The reason is that TIGeR tries to select more reliable routes by using traffic information which leads to better delivery ratio, specially in sparse networks. Moreover, as it is expected, TIGeR+RR makes higher average path length than TIGeR-RR. This is because of returning the packets which face local optimums (e.g., a dead-end road or in case of network partitioning). TIGeR+RR uses route recovery to return these packets to the last intersection node to select possible alternative roads. This means increase in path length.

End-to-end delay is considered as the time distance between generating a data packet and the time it reaches its destination. Simulation results show that TIGeR-RR has an average end-to-end delay very close to that of GPSR and AODV for different network densities. It means that higher average path length does not necessarily lead to higher latency. Note that we are only able to calculate the latency for the delivered packets to the destination. TIGeR provides higher delivery ratio and the calculated average latency is also for those packets that AODV or GPSR cannot deliver them. Accordingly, the average latency using TIGeR+RR is higher than that of TIGeR-RR, in particular in sparse networks. This increase in delay is only due to the packets that are using carry and forward and route recovery mechanism and TIGeR-RR is not able to deliver them to their destination. Thus the average end-to-end delay of TIGeR+RR for those packets that do not use carry and forward to reach the destination is very close to that of GPSR. For dense networks with better wireless connectivity, carry and forward is much less necessary and thus TIGeR+RR provides an overall end-to-end delay very close to that of TIGeR-RR.

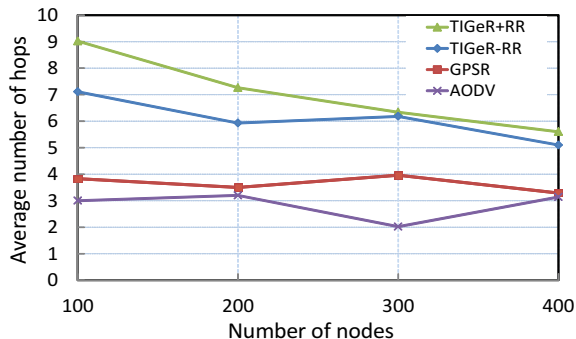
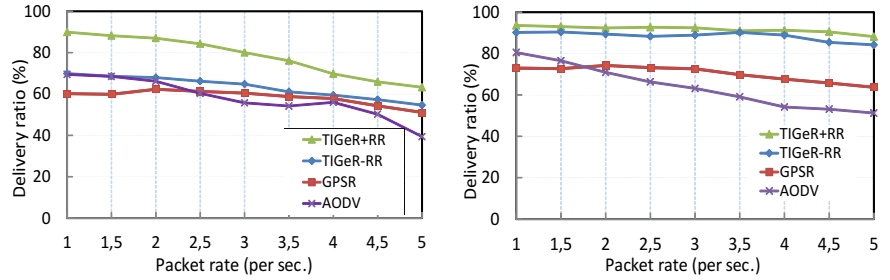


Fig. 5. The average number of hops traversed by packets.



(a) PDR for number of node = 100.

(b) PDR for number of node = 400.

Fig. 4. Achieved PDR for different rates of packet generation by the source nodes.

## V. CONCLUSIONS

This paper proposes TIGeR, a new geographical routing protocol for urban environments. This protocol uses real-time local traffic information at intersections to improve the performance of routing by selecting reliable road-based routes. The protocol is fortified by a neighborhood-based intersection detection mechanism which relieve the protocol from the need for a road map. To forward packets along the roads between adjacent intersections, an angular forwarding method is exploited. TIGeR also includes a carry and forward method and route recovery strategy to deal with the local optimum problem such as dead-end streets. Simulation results show that TIGeR outperforms other protocols (GPSR and AODV) in term of packet delivery ratio without a negative impact on the latency and network overhead.

As future work, we are implementing other VANET routing protocols to compare with TIGeR. Taking data load of nodes into account for making routing decisions for load balancing is also considered as our future work.

## REFERENCES

- [1] D. B. Johnson and D. A. Maltz, "DSR: Dynamic source routing in ad hoc wireless networks," in *Proc. Mobile Computing*, 1996.
- [2] C. E. Perkins et al., "AODV: Ad-hoc on-demand distance vector routing," in *Proc. Mobile Computing Systems and Applications*, 1999.
- [3] B. Karp and H.-T. Kung, "GPSR: greedy perimeter stateless routing for wireless networks," in *Proc. of the ACM/IEEE Conf. on Mobile Computing and Networking (MobiCom)*, ACM, 2000.
- [4] C. Lochert et al., "GPCR: Geographic routing in city scenario," *ACM SIGMOBILE Mobile Computing and Communications Review (MC2R)*, vol. 9, 2005.
- [5] B.-C. Seet et al., "A-STAR: A mobile ad hoc routing strategy for metropolis vehicular communications," in *Proc. NETWORKING*, 2004, pp. 1989-999.
- [6] C. Lochert et al., "GSR: A routing strategy for vehicular ad hoc networks in city environments," in *Proc. Intelligent Vehicles Symposium*. IEEE, 2003.
- [7] J. Zhao et al., "VADD: Vehicle-assisted data delivery in vehicular ad hoc networks," in *Proc. Conf. on Computer Communications*. IEEE, 2006.
- [8] Y. Ding and L. Xiao, "SADV: Static-node-assisted adaptive data dissemination in vehicular networks," *IEEE Transactions on Vehicular Technology*, vol. 59, pp. 2445-2455, 2010.
- [9] M. Jerbi et al., "GyTAR: Towards efficient geographic routing in urban vehicular networks," *IEEE Transactions on Vehicular Technology*, vol. 58, pp. 5048-5059, 2009.
- [10] "The ns-2 network simulator. available: <http://www.isi.edu/nsnam/ns/>."
- [11] "Hierarchical location service (HLS)+GPSR. <http://www.cn.uni-duesseldorf.de/alumni/kiess/software/hls-ns2-patch>, December, 2012."
- [12] "U.s.census bureau-TIGER/Line 2006 second ed. available: <http://www.census.gov/geo/www/tiger/>."
- [13] M. Pirkowski et al., "TraNS: Realistic joint traffic and network simulator for VANETs," in *ACM SIGMOBILE Mobile Computing and Communications Review*. ACM, 2008.
- [14] "SUMO-simulation of urban mobility, <http://sumo.sourceforge.net/>."