

UTCARP: URBAN TRAFFIC CONTROL AWARE ROUTING PROTOCOL

Hekmat Mohammadzadeh¹ and Shapour Joudi Bigdello²

^{1,2}Department of Computer, Parsabad Moghan Branch, Azad University, Parsabad, Iran

¹hekmat@iaupmogan.ac.ir

²joudi@iaupmogan.ac.ir

ABSTRACT

Vehicular Ad hoc networks (VANETs) are being advocated as a means to increase road safety and driving comfort, as well as to facilitate traffic control. Road congestion and traffic-related pollution have a large negative social and economic impact on several economies worldwide. Due to the high dynamic nature of the network topology in VANETs, finding and maintaining the routes for data forwarding is still more challenging. In this paper, we propose a urban traffic control aware routing protocol for VANETs that is called UTCARP. It considers two modules of (i) the traffic control aware selection of vertices through which a packet is passed toward its destination and (ii) the greedy forwarding strategy by which a packet is forwarded between two adjacent vertices. The simulation results illustrate that the proposed approach outperforms conventional protocols in terms of packet delivery ratio, end-to-end delay and routing overhead.

KEYWORDS

VANET, UTCARP, Traffic Control, Routing Protocol.

1. INTRODUCTION

Vehicular Ad hoc Networks (VANETs) are a special type of Mobile Ad hoc Networks (MANETs), made by vehicles communicating among themselves, and by vehicles communicating to devices located in the margins of roads and highways [8][9][10]. VANETs [8] are emerging as new infrastructures for monitoring the physical world, especially in urban areas where a high concentration of vehicles equipped with onboard device is expected. VANETs provide a perfect way to collect dynamic interest of information. Figure 1 is an application example of VANETs. VANETs sensors are mounted on the vehicles, and each sensor captures the urban physical quantities, e.g. urban temperature, and then forwards them to the nearest base station. Then, the base station can send the information to the application server by using one or more wired networks.

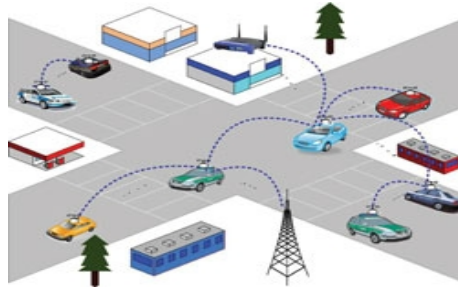


Figure 1. Example of VANET in city environment

In fact, vehicular sensors are not affected by strict energy constraints and storage capabilities because they can be equipped with powerful processing units and wireless transmitters in a vehicle. Consequently, energy dissipation and data storage space are not considered often as design issues of routing protocols in VANETs. The major routing issue considered in VANETs is the rapidly changing network topology. This is because wireless communication is unreliable in vehicle-to-vehicle (V2V) communication due to multipath fading, shadowing, and Doppler effects caused by the high mobility of vehicles. Such effects make routing protocols quite complicated. In this paper, we aim to design a routing protocol based on geographic forwarding for VANET, which is associated with the traffic control mechanism. The vertices selection and the greedy forwarding between neighbouring vertices are based on the current traffic situation. Several assumptions have been made in the paper as follows:

- Vehicles participating in a VANET can know their own position, speed, moving direction and acceleration/deceleration correctly by using the electronic control unit and navigation system.
- Vehicles are equipped with identical pre-loaded urban digital maps.
- Vehicular sensors have plentiful space for storage and power supply. The energy dissipation and storage usage are not taken into account in this paper.
- Vehicular sensors sense and recognize the physical quantities correctly.

The remainder of this paper is organized as follows: Section 2 overviews two related routing strategies based on geographic forwarding for urban wireless vehicular networks. The system model of network and mobility is outlined in section 3. In section 4, the proposed UTCARP is presented. Section 5 discusses the performance evaluation of UTCARP. The simulation environment and results are presented in detail. Finally, the paper is concluded in section 6.

2. RELATED WORK

In this section, we look at the existing routing proposals in VANET and then discuss the inconvenience of using such protocols in the vehicular environment, especially in the city environments.

2.1. Routing in VANET

Recently, some routing protocols specific to VANETs have been proposed. In the following, we present the most important ones: GSR, GyTAR, A-STAR, and GPCR.

Geographic Source Routing (GSR) [2] uses reactive location service (RLS) to know the current position of the desired communication partner. When the querying node requires position

information of neighbouring nodes, it floods the "position request" containing its ID to the network in a reactive way. When the corresponding node receives the request, it sends "position reply" to the querying node. With the position information of neighbour nodes, the sender node computes a sequence of junctions, through which a packet has to traverse to reach its destination using a city map. Note that the sequence of junctions can be either contained in the packet header or computed by each forwarding node. Forwarding a packet to successive junctions is done based on greedy forwarding and using the Dijkstra's shortest path algorithm [13], and the distance from source to destination can be calculated based on the city map. When a route break occurs, GSR uses the recovery strategy "fall back on greedy mode" to bypass the particular node.

Improved Greedy Traffic Aware Routing Protocol (GyTAR) [4] is an improved greedy traffic aware, intersection-based geographic routing protocol that uses real-time traffic density information and movement prediction to route packets. It consists of two modules of (i) selection of junctions through which a packet must pass to reach its destination and (ii) an improved greedy forwarding mechanism between two junctions [4]. When a vehicle receives a packet, it computes its next junction with the highest score by considering traffic density and curve-metric distance to the destination. The junction with the highest score is geographically closest to the destination vehicle and has the highest vehicular traffic. Between two adjacent junctions, the packets are forwarded through the vehicles on between the successive junctions by using improved greedy forwarding. Each vehicle maintains a table containing position, velocity and direction of each neighbouring vehicle, and the table is updated by periodically exchanging HELLO messages among vehicles. Using the information in the table, forwarding vehicles select their next hop neighbour that is closest to the destination junction.

A-STAR [3] (Anchor-based Street and Traffic Aware Routing) is a position-based routing scheme designed specifically for IVC in a city environments. It features the novel use of city bus route information to identify anchor paths of higher connectivity so that more packets can be delivered to their destinations successfully. A new recovery strategy for packets routed to a local optimum was also proposed, consisting of the computation of a new anchor path from the local maximum to which the packet is routed.

The Greedy Perimeter Coordinator Routing (GPCR) protocol [8] has been designed to deal with the challenges of city scenarios. It does not require any global or external information such as a static street map. The main idea of GPCR is to forward data packets using a restricted greedy forwarding procedure. That means when choosing the next hop, a coordinator node (a node on a junction) is preferred to a non-coordinator node, even if it is not the closest node to destination.

2.2. Discussion

In the previous sections, we discussed VANET characteristics including high-speed node movement, frequent topology change, and short connection lifetime especially with multi-hop paths. These three characteristics degrade significantly the performance of conventional topology based routing protocols designed for MANETs. This is due to packet control overhead (route discovery, route maintenance, etc.) caused by frequent update of routing information of the whole network, route failures and transient nature of links. The frequently changed topology suggests that a local routing scheme without the need to keep track of global routing information scales better in VANET and consume a low wireless bandwidth. In addition, the popularity of the Global Positioning System (GPS) also makes position-based routing, which maintains only local information about the node's position, a popular routing strategy. However, at the same time, the direct application of geographic routing protocols to VANET is not suitable. Indeed, we note that existing geographic routing improvements are often based on a simple greedy forwarding concept (closest vehicle to the destination) without taking into account urban

environment characteristics. This leads to difficult signal reception due to radio obstacles such as high-rise buildings. The proposed vehicular routing protocols solved this problem by forwarding packets through the sequence of vertices. However, applying intersection-based routing to IVC may not be without any problems. An example is GSR [2], where the sender calculates the shortest path to the destination using the Dijkstra algorithm and according to the street map. Then it computes a sequence of junctions through which the packet has to pass in order to reach the destination. Note that this approach does not take into account the vehicular traffic. That means the next street to be taken is determined without considering whether there is sufficient number of nodes on the street. A-STAR [3] also suffers from problem of connectivity on some sections of streets since it uses static vehicular traffic information based on city bus routes to find a path from source to destination. Moreover, in A-STAR and GSR, forwarding a packet between two successive junctions is done based on simple greedy forwarding mechanism without considering vehicle direction, velocity. Thus, the selected vehicle chosen to forward data packet might not be the best choice.

3. SYSTEM MODEL

In this section, we look at the network model and mobility model in the vehicular environment, especially in the city environments.

3.1. Network Model

In this paper, it is assumed that all vehicles communicate with each other by using IEEE 802.11 standard. In city environments, high-rise buildings are the radio obstacles. In Figure 2, vehicle B is within the communication range of vehicle A. Vehicle A forwards a packet to vehicle B, but vehicle B cannot receive the packet from vehicle A because of radio obstacles. In such an area, while greedy forwarding is used to forward a packet to its neighbour, source node (node and vehicle are used interchangeably) chooses a neighbour which is closest to the destination node within its communication range but the transmitted packet may be lost due to radio obstacles.

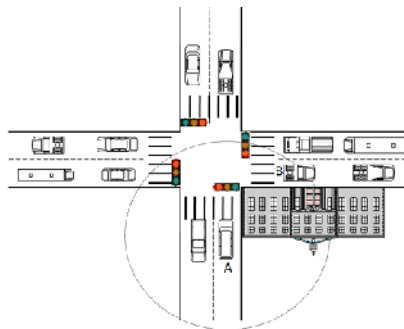


Figure 2. Road obstacles problem in city environment

3.2. Mobility Model

In VANETs, the mobility pattern of vehicles influences the route discovery, maintenance, reconstruction, and accuracy [1]. We illustrate the three key factors of street layout, obstacles, and traffic control mechanism that affect the mobility of vehicles as follows:

1. *Street layouts*: Streets force vehicles to confine their movement to well-defined paths. This constrained movement pattern determines the spatial distribution of vehicles and their connectivity. Streets can have either single or multiple lanes and can allow

either one-way or two-way traffic. The movement of every vehicle is influenced by the movement pattern of its surrounding vehicles. For example, a vehicle would try to maintain a minimum distance from the one in front of it. It may increase or decrease its speed, or may change to another lane.

2. *Obstacles*: Obstacles, such as buildings, determine the number of intersections in the area, which in turn determines the frequency of vehicle stops. They also determine whether vehicles at the neighboring intersections can sense each other's radio transmissions. Larger obstacles make the network more sensitive to clustering and degrade performance.
3. *Traffic control mechanisms*: In this paper, we assume every street intersection has a traffic light. If vehicles following each other move to an intersection with a red light, the vehicles form a queue at the intersection. Each vehicle waits for at least the required time once it gets to the head of the intersection after other vehicles ahead in the queue leave. The traffic light gives the vehicles a probability, denoted as P_{inter} , to stop at the intersection when the vehicles reach it with an empty queue. With the probability $1 - P_{inter}$, the vehicles can directly cross the intersection without stopping. In UTCARP, for every street intersection, we use a unique P_{inter} for vehicles stopping at intersections with an empty queue because traffic lights are altered periodically after the implementation of the system. At the same time, we set a stop sign in the middle of the each street segment. If a vehicle moves to a stop sign with an empty queued vehicles line, it stops at the stop sign with a probability P_{st} . With the probability $1 - P_{st}$, the vehicle can pass by the stop sign immediately. The value of P_{st} for different street segments, however, varies because it is determined by the roadside buildings, such as schools, hospitals and restaurants. Obviously, a vehicle moving on a street with more roadside objects has a higher value of P_{st} . We assume that the vehicles that move on the same street segment can share a unique P_{st} . However, the value of P_{st} varies between street segments (An example of the distribution of P_{st} will be given in Section 5). Furthermore, if a vehicle decides to wait in an empty queue, the amount of waiting time is randomly chosen between zero and T seconds. Any vehicle that arrives later at a non-empty queue will have to wait for the remaining wait time of the previous vehicle plus one second. The additional one second simulates the start up delay between queued vehicles. Whenever the traffic light or stop sign turns green, the vehicles begin to cross the signal at intervals of one second, until the queue is empty. The next vehicle that arrives at the head of an empty queue again makes a decision on whether to stop with a probability P_{inter} or P_{st} and so on.

In the system, Manhattan mobility [5] is used for vehicles that move in a grid road topology mainly proposed for movement in an urban area, where the streets are organized in a regular grid. In this mobility model, the mobile vehicles move in horizontal or vertical directions on an urban map.

4. PROPOSED PROTOCOL: UTCARP

In this section, we describe proposed protocol. Most conventional V2V routing protocols do not consider the factors affecting the vehicles' mobility. However, the mobility models determine the location of nodes in the topology at any given time interval and they strongly affect network connectivity and throughput. In this paper, we aim to design a new V2V routing protocol for VANETs associated with urban traffic control mechanism, which inevitably influence vehicles mobility. The proposed UTCARP is divided into two phases: (i) vertices selection and (ii) packet forwarding between two adjacent vertices. They are detailed as follows.

4.1. Vertices Selection

As a strategy to deal with the high mobility of nodes on one hand and with the specific topological structure of a city on the other hand, we have chosen a position-based routing protocol, which is supported by a digital map of the city. The presence of a digital map is a valid assumption when vehicles are equipped with on-board navigation systems. Thus, each vehicle is aware of its geographic position, and knows the position of neighbors by sensing beacon messages that are periodically exchanged by vehicles and roadside infrastructure. We also assume that every vehicle is aware of the current traffic status. This information can be provided through a simple distributed mechanism for on-road traffic estimation realized either by all vehicles or by traffic sensors installed beside the streets.

In GSR [2], the packet-sending node can compute a path to the destination by using the navigation system. This path can be abstracted as a directed graph $P(V,E)$ where V is the set of vertices and E is the set of edges. The sequence of vertices can be put into the packet header, and forwarding the packet between two successive vertices is done based on greedy forwarding. The path between source and destination in GSR is determined by the Dijkstra shortest path calculation based on the street map. In Figure 3, upon sensing any event on the road, the sender, vehicle A, communicates with the nearest base station (sink) in an ad hoc manner among local vehicles. An example of the shortest path determined by the Dijkstra algorithm is vehicle A-V1-V2-V4-V6-BS. UTCARP, similar to conventional position-based source routing protocols, adopts the anchor-based routing approach with street awareness. Therefore, data packets will be route between vehicles according to the street map topology. However, unlike the conventional V2V routing protocols, where the sender statically computes a sequence of vertices the packet has to traverse in order to reach the destination, intermediate vertices in UTCARP are chosen dynamically one by one, considering both distance to destination and the current traffic status. Each data-forwarding vehicle associated with the periodically updated traffic information determines involved intermediate vertices.

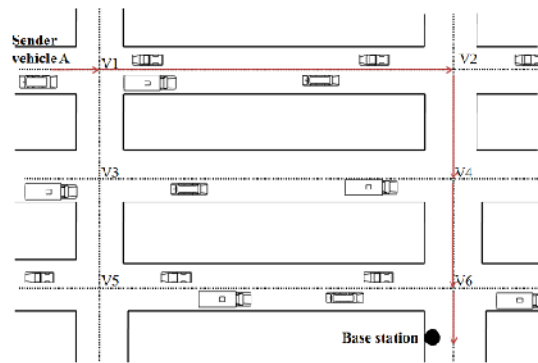


Figure 3. Example of routing path in GSR

Figure 4 exhibits the problem of conventional routing protocols without considering traffic control mechanism. According to Figure 3, the calculated shortest routing path is A-V1-V2-V4-V6-BS. However, as illustrated in Figure 4, if the sender stops and clusters with its neighbours at the street intersection or moves in a different direction from the computed shortest packet forwarding path, there may be no vehicles which can be the next router to forward the packet along edge $E(V1,V2)$. In this case, by considering the distance between the source and destination, a substitute routing path is $E(V1,V3)$, where the packet can be forwarded to vehicle A, and then forwarded to vehicle B.

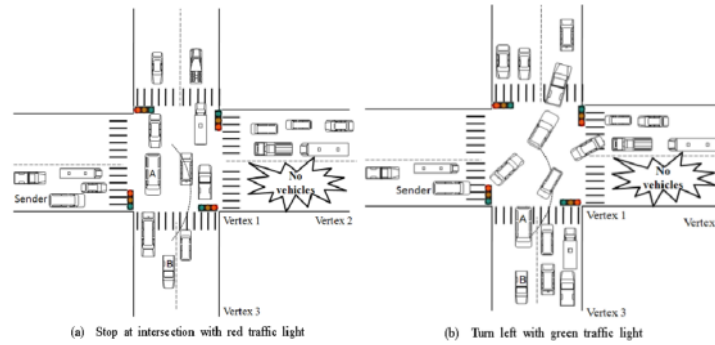


Figure 4. Example of traffic status

In Figure 3, let us suppose that the sender wishes to forward data to the nearest base station. It can identify a sequence of vertices between itself and the nearest base station with the help of a city map data provided by the navigation system. There are three identified routes to get to the destination; they are A-V1-V2-V4-V6-BS, A-V1-V3-V4-V6-BS and A-V1-V3-V5-V6-BS. We assume that the sender prefers the shortest path among these three routes. However, a routing hole problem occurs at V1 as illustrated in Figure 4. The sender A identifies this situation and then reselects the third path to forward the packet. Additionally, if the packet is relayed at V5, and there is only one path E(V5,V6) can minimize $dis(S,D)$. In this case, we use the strategy of "carry and forward" [6] to send the packet close to the destination when there is no forwarding vehicle on the calculated shortest path.

The prediction of a sequence of vertices in UTCARP is done as follows:

Notations:

- S*: Sender.
 - D*: Destination.
 - $dis(S, D)$: Distance between *S* and *D*.
 - N*: Next forwarding node of current sender.
 - Tab*: Computed routing table with the shortest routing path.
 - i*: Current vertex.
 - j*: Next vertex of *i* along with the shortest routing path computed by the sender.
 - $V_{(1,2,3...n...k)}$: The set of other neighbor vertices of *i* except *j*.
- Note here, if we use the two-street intersections, $k = 2$.
-

Algorithm 1: Vertex selection

- 1: **if** a packet is relayed from the previous upstream node to *i*
 - 2: **if** there is no *N* within the transmission range of *S* in the direction of *j*
 - 3: **if** there is V_n can minimize $dis(S,D)$ **then**
 update *Tab* and forward the packet to V_n with the minimum $dis(S,D)$ according to the greedy forwarding mechanism.
 - 4: **else**
 carry and forward the packet to *j*
 - 5: **else**
 forward the packet to *j* according to the greedy forwarding mechanism
 - 6: **else**
 forward the packet to *i* according to the greedy forwarding mechanism
 - 7: **end if**
-

4.1. Packet Forwarding Between Tow Vertices

Once the sequence of valid vertices between the source vehicle and base station is determined, the improved greedy strategy is used to forward packets between the two involved vertices. Each vehicle maintains a neighbour table in which the position, velocity, acceleration/deceleration and direction of each neighbour vehicle are recorded. This table is updated through hello messages exchanged periodically by all vehicles. Thus, when a packet is received, the forwarding vehicle computes the new predicted position of each neighbour using the recorded information and then selects the next hop neighbour. We explain the proposed greedy routing strategy based on Figure 5. In Figure 5(a), when vehicle A is moving in the same direction as the sender with a higher speed than vehicle B, vehicle A will receive the forwarded packet since at time 2 illustrated in Figure 5(b), it is the closest vehicle to the next vertex. Without using this prediction, the forwarding vehicle would choose vehicle C as leading the routing loops. In this paper, as we implement the stop signs beside the streets as illustrated in Figure 5(c) and (d), the greedy routing prediction will become more complicated. In Figure 5(c), if vehicle A has a higher moving speed it is supposed to receive the packet from the sender. However, if vehicle A has to stop at a stop sign with probability of P_{st} as illustrated in Figure 5(d), vehicle B then, is better than vehicle A to be the forwarding router, and will receive the packet from the sender. In this case, we cannot determine the forwarding vehicle by the vehicle's direction and speed. We also need to consider the vehicle's position and acceleration/deceleration. It is obvious vehicle A has the highest deceleration, which is not an ideal next forwarding vehicle even if it moves faster than vehicle B. This is because vehicle A will stop somewhere in a short time, due to the environmental constraints. In contrast, vehicle B has already passed the stop sign, and can move at a stable speed, without deceleration. Consequently, vehicle B is the ideal forwarding neighbour for the sender vehicle. In this situation, there is a high risk that a packet will be stuck in a local optimum, where the forwarding vehicle might be the closest to the next vertex. Hence, a recovery strategy is required. The repair strategy of UTCARP is based on a "carry and forward" scheme [6], where the forwarding vehicle of the packet in recovery mode will carry another vehicle, closer to the destination.

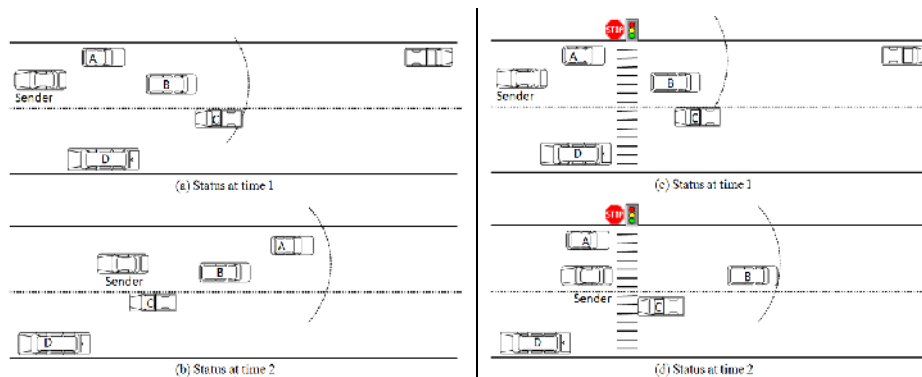


Figure 5. Example of greedy forwarding

5. PERFORMANCE EVALUATION

In this section, we evaluate the performance of our proposed routing protocol by using the NS2.34 simulator [7]. We compare the performance of UTCARP with existing routing protocols GSR [2] and GyTAR [4]. They are representative geographic routing protocols, for V2V wireless networks.

5.1. Simulation Environment

The experiment is based on a 2000m×2000m rectangular street area, which presents a grid layout. For the simulation, a 2000m×2000m area is chosen, consisting of 25 junctions or intersections and 10 two-way roads. This street layout is derived and normalized into a realistic mobility trace from a Manhattan mobility model. The map data was then transformed into the data format used by the NS2, simulation tool. Vehicles with random start points and destinations were placed on the map. The model vehicles were assigned a maximal speed of 60km/h with accelerating/decelerating speeds of -10~10 m/s². Each vehicle had radio propagation ranges of 250m. For the performance evaluation, 15 random connections were established using CBR traffic varying 1~16 packet(s)/second with a packets size of 128 bytes. The value of the probability Pinter to stop the vehicles at street intersections when the vehicles reach an empty queue was set to 0.25. On the other hand, the probability Pst that a vehicle would stop at a sign with an empty queue was randomly set in a range from 0.1 to 0.5. An example of distributed Pst of each stop sign is shown in Figure 6.

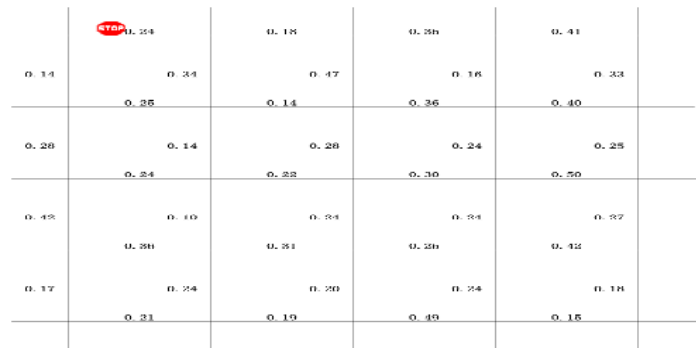


Figure 6. Example of distribution of Pst

Take note that we varied the value of distributed Pst in different simulation runs. The maximum value T for waiting at intersections or stop signs is given as 10 seconds. The simulation results are averaged over ten runs. All the key parameters of our simulation are summarized in Table1. Each simulation takes 900 second of simulation time. The performance metrics used to evaluate the simulation results are as following:

- Packet delivery ratio: the ratio of originated data packets that are successfully delivered to their destinations to the original sent ones.
- Average end-to-end delay: the average time it takes for a packet to traverse the network from its source to destination.
- Routing overhead: the ratio of the total number of bytes of control packets to the total number of bytes of data packets delivered to the destinations during the entire simulation.

The routing protocols are compared under various data transmission rates and various vehicle densities. For the traffic generation in variable node densities, we set a constant packet sending rate i.e., 4 packets/second. On the other hand for traffic generation with variable packet sending rate we kept the number of nodes constant i.e., 200 nodes. Detailed analysis of the simulation results are given in the following.

Table 1. The parameters of used in simulation

Simulation setup/Scenario		MAC/Routing	
Network area	2000m × 2000m	Channel capacity	2 Mbps
Filter	0-25	Packet sending rate	1-16 packets/second
Psi	0.1-0.5		
Vehicle speed	0-60 km/hour	Data packet size	128 bytes
Acceleration	-10~10 m/s ²		
Deceleration	-10~10 m/s ²		
Stop time (T)	10 seconds	MAC protocol	IEEE 802.11 DCF

5.2. Simulation Result and Discussion

1. *Packet delivery rate:* In this part, we compare the performance of UTCARP, GSR, and GyTAR in terms of packet delivery. For better performance, protocols should be tolerable to a small amount of packet loss. We will show how packet delivery is affected by the packet send rate and the nodes' density. In Figure 7(a), GSR has the worst performance, i.e. less than 50% delivery ratio for 16pkt/sec. In case of GyTAR, delivery rates increase up to almost 56% for 16pkt/sec. Our proposed UTCARP achieves the highest packet delivery ratio across all packet send rates observed. As many as 10% more packets are delivered by UTCARP than GyTAR. This is mainly because in UTCARP, the path is determined progressively following the current road traffic status. The data routing path is altered when routing holes occur due to traffic control mechanisms (while only the shortest path is used for route selection in GSR and the path with the most nodes is selected in GyTAR). A packet will move successively closer to the destination along streets, which have good traffic situations providing good network connectivity. In Figure 7(b), all three protocols improve in reliability as the number of nodes increased. This is expected since more nodes increases the probability of connectivity, which in turn reduces the number of packets dropped due to local maximums. However, when the network density increases too much (>200) there is a decrease in the delivery ratios of GSR and GyTAR. This is because there is a high probability of vehicles being queued in front of stop signs and street intersections. Radio interference and collisions between nodes increase when many nodes are clustered together (UTCARP can improve the delivery ratio decrease threshold value up to 250). In this situation, we need a traffic status awareness routing protocol, which selects the routing path based on the current traffic status. In general, UTCARP has a much higher delivery ratio than competitors because with local traffic awareness the packets can be routed successfully instead of being dropped.

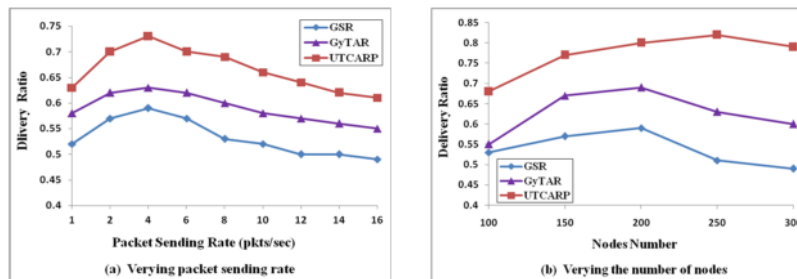


Figure 7. Packet delivery rate

2. *Average end-to-end delay:* In this section, we compare the performance of UTCARP with GSR and GyTAR in terms of end-to-end delay. As shown in Figure 8, UTCARP achieves a much lower end-to-end delay than GSR and GyTAR in all tested configurations. This is mainly because in UTCARP, the number of hops involved to deliver packets is reduced due to the improved greedy strategy used to forward packets between two connected vertices, and also because UTCARP does not need to keep track of an end-to-end route before sending data packets from source to destination. More importantly, UTCARP not only considers the moving direction and speed of neighbouring vehicles but also considers the position and acceleration/deceleration of them. This condition can help our protocol to choose a stable route for forwarding data to the destination. Delay in GSR is higher than GyTAR and UTCARP because packets whose deliveries were suspended are stored in the buffer for a longer time than in GyTAR and UTCARP's. GyTAR's delay is higher than UTCARP because GyTAR is not suited for more complicated traffic environments, and will select nodes with high moving speeds but low deceleration unlike UTCARP. Figure 8(a) shows the results of varying packet send rates. Up to packet sending rate 4, the three delay plots decrease slightly, but after that point, they start slightly increasing. Figure 8(b) illustrates the results of varying the node numbers. The plots display the opposite trend of delivery ratio. It first decreases as the number of nodes increases, and then (up to 200 for GSR and GyTAR, 250 for UTCARP) there is an increase thereafter.

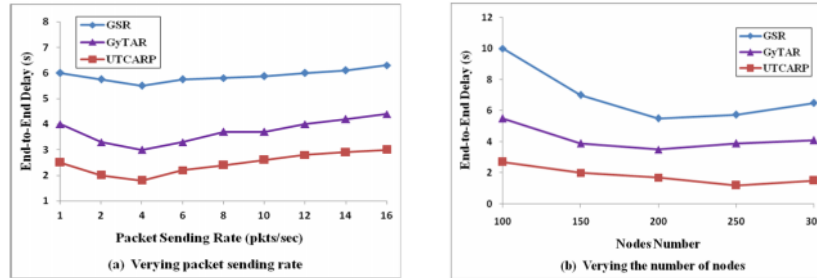


Figure 8. End-to-end delay

3. *Routing overhead:* In Figure 9(a), it is observed that routing overhead decreases for all the protocols while increasing the packet sending rates up to 4 packets/sec. However, beyond four packets/sec routing overhead remains almost constant in all the routing protocols. This is expected since the number of control messages is constant for the same number of nodes (i.e. number of nodes is set to 200). As shown in Figure 9(b), an increase in vehicle density leads to an increase in routing overhead since the rate of control messages depends on the number of nodes. In general, UTCARP outperforms the two competitors in all cases of varying data transmission rates and with different vehicle densities. This is because in UTCARP, as in GyTAR, we have only three types of control messages, including Route request, Route reply, and Route error, which are used for route discovery and route maintenance. These control messages are updated accurately with the current traffic status. Therefore, in UTCARP, there is a low message re-transmitting rate that yields a low overhead plot. Although GSR uses only hello messages as control messages, it shows a higher routing overhead than UTCARP and GyTAR. This is because UTCARP and GyTAR do not need as many hello messages sent as GSR. This is due to the mechanism for a neighbour's position inference used in UTCARP and GyTAR. Hence, the frequency of hello messages needed by GSR is more than UTCARP and GyTAR.

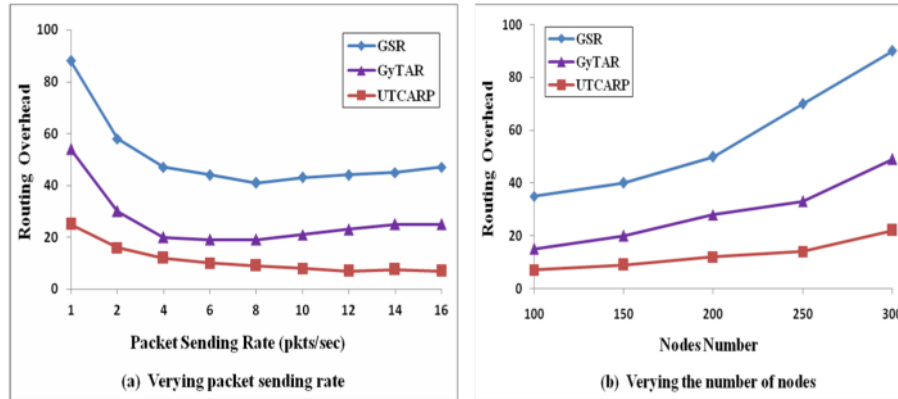


Figure 9. Routing overhead

6. CONCLUSIONS

In this paper, a routing protocol based on geographic forwarding has been proposed for vehicular ad hoc networks in an urban environment, which is associated with the traffic control mechanism that is called UTCARP. It considers the traffic control mechanism of traffic lights and stop signs. UTCARP performs the two key operations of the prediction of a sequence of vertices and the use of the predictive directional greedy routing to forward the data from a source vehicle to a destination through the sequence of vertices. The simulation results illustrate that the proposed approach outperforms conventional protocols in terms of packet delivery ratio, end-to-end delay and routing overhead.

REFERENCES

- [1] A. K. Saha and D. B. Johnson, "Modeling Mobility for Vehicular Ad Hoc Networks", Appeared as a poster in the First ACM Workshop on Vehicular Ad Hoc Networks (VANET 2004), Philadelphia, Pennsylvania, October 2004.
- [2] C. Lochert, H. Hartenstein, J. Tian, D. Herrmann and M. Mauve, "A routing strategy for vehicular ad hoc networks in city environments," Proc. of IEEE Intelligent Vehicles Symposium (IV2003), pp. 156-161, June 2003.
- [3] B.-C. Seet, G. Liu, B.-S. Lee, C. H. Foh, K. J. Wong, K.-K. Lee, "A-STAR: A Mobile Ad Hoc Routing Strategy for Metropolis Vehicular Communications". IFIP NETWORKING, pp. 989-999, Athens, Greece, December 2004.
- [4] M. Jebri, S. Mohammed, R. Meraihi, and Y. G. Doudane, "An Improved Vehicular Ad Hoc Routing Protocol for City Environments," Proc. of IEEE Int. Conf. on Communications, pp.3972-3979, Jun. 2007.
- [5] B. Divecha, A. Abraham, C. Grosan and S. Sanyal, "Impact of Node Mobility on MANET Routing Protocol Models." Available at <http://scholar.google.com>, 2010.
- [6] J. Li, J. Jannotti, D. De Couto, D. Karger, and R. Morris, "A scalable location service for geographic ad hoc routing", Proc. of ACM/IEEE MOBICOM'2000, pp. 120-130, 2000.
- [7] F. A. I. W. on Vehicular Ad Hoc Networks (VANET), "Fleetnet: communication platform for vehicular ad hoc networks," in Zukunftsforum Mobiles Internet, October 2004.
- [8] C. Lochert, H. Hartenstein, J. Tian, D. Herrmann, H. Füßler, M. Mauve, "A Routing Strategy for Vehicular Ad Hoc Networks in City Environments", IEEE Intelligent Vehicles Symposium, pp. 156-16, Columbus, OH, USA, June 2003.
- [9] C. Lochert, M. Mauve, H. Füßler, H. Hartenstein, "Geographic Routing in City Scenarios", ACM/IEEE MOBICOM'2004, Philadelphia, PA, USA, September 2004.
- [10] Z. Chen, H. Kung, and D. Vlah, "Ad Hoc Relay Wireless Networks over Moving Vehicles on Highways", Poster, ACM Mobihoc'2001, pp. 247-250, Long Beach, CA, October 2001.

- [11] H. Fbler, M. Mauve, H. Hartenstein, M. Kemann, and D. Vollmer, "Location-based routing for vehicular ad hoc networks", student poster, ACM/IEEE MOBICOM, Atlanta, Georgia, USA, September 2002.
- [12] Car2Car Communication Consortium, www.car-to-car.org.
- [13] Dijkstra's algorithm. Available at: http://en.wikipedia.org/wiki/Dijkstra's_algorithm.
- [14] The Network Simulator NS-2, Available at: <http://www.isi.edu/nsnam/ns>, 2010.

Authors

Hekmat Mohammadzadeh received the M.Sc. degree in computer engineering from Azad University, Tabriz, Iran, in 2010. He is ICT manager and faculty member in department of computer engineering at Azad University of Parsabad Moghan Branch. His research interest includes wireless network, MANET and VANET.



Shapour Joudi begdillo received the M.Sc. degree in computer engineering from Islamic Azad University, Qazvin, Iran, in 2007. He is faculty member in department of computer engineering at Azad University of Parsabad Moghan Branch. His research interests are wireless network and distributed systems

