

MURU: A Multi-Hop Routing Protocol for Urban Vehicular Ad Hoc Networks

Zhaomin Mo, Hao Zhu, Kia Makki, Niki Pissinou
Telecommunications and Information Technology Institute
Florida International University
Miami, FL 33174
E-mail: {zmo002, hao.zhu, makkik, pissinou}@fiu.edu

Abstract—Vehicular ad hoc networks (VANETs) are going to be an important communication infrastructure in our life. Because of high mobility and frequent link disconnection, it becomes quite challenging to establish a robust multi-hop path that helps packet delivery from the source to the destination. This paper presents a multi-hop routing protocol, called MURU, that is able to find robust paths in urban VANETs to achieve high end-to-end packet delivery ratio with low overhead. MURU tries to minimize the probability of path breakage by exploiting mobility information of each vehicle in VANETs. A new metric called *expected disconnection degree* (EDD) is used to select the most robust path from the source to the destination. MURU is fully distributed and does not incur much overhead, which makes MURU highly scalable for VANETs. The design is sufficiently justified through theoretical analysis and the protocol is evaluated with extensive simulations. Simulation results demonstrate that MURU significantly outperforms existing ad hoc routing protocols in terms of packet delivery ratio, packet delay and control overhead.

I. INTRODUCTION

Recently, there has been increasing interest in exploring computation and communication capabilities in transportation systems ([1],[10]). Many automobile manufactures started to equip GPS, digital map and communication interfaces (e.g. 802.11 WLAN cards) with new vehicles. Existing cars can also be easily upgraded. With the rapid advance of information technology, it becomes easy to support low-cost inter-vehicle communication, and then to provide customized service to individual drivers ([18],[5]). One example of service is to disseminate traffic information to drivers in a certain area, which helps drivers to choose the fast route to their destinations by making a detour of jamming roads. Another example is to let drivers share entertainment files such as funny flashes or movie clips to make the trip enjoyable.

There are several candidate network architectures for intelligent transport systems. One is that vehicles use cellular network infrastructure for communication. This architecture may have two drawbacks, which are high operation cost and limited bandwidth. Another architecture is a hybrid one that combines the vehicle-to-vehicle communication and vehicle-to-base station communication to alleviate the tight bandwidth budget of cellular networks. This architecture still requires that each vehicle subscribes data service in cellular networks, whose cost may be still quite high. Moreover, when vehicles

reside in the area where cellular networks are not available or damaged, it would be difficult to keep smooth inter-vehicle communication. As a result, people have paid a lot of attention to vehicle ad hoc communication architecture, which greatly increases the flexibility of deployment and reduces the cost as well. Without the need of fixed infrastructure, vehicle with wireless network interfaces (e.g. WiFi interface) can spontaneously establish a vehicular ad hoc network (VANET) anytime anywhere.

In VANETs, each vehicle or node¹ moves on the road with high speed and the trajectory of the vehicle can usually be predicted by itself since the mobility pattern (e.g. direction and the speed limit) of the vehicle can be approximately inferred by the roadway geometry. Because nodes move with high velocity in VANETs, and the channel condition of each link is highly error-prone due to the reflection of tall buildings and obstacles along the road and interference from other sources (e.g. WiFi hotspots), the network topology of VANETs could be highly dynamic, which means frequent link disconnection may happen. This makes it challenging to set up a robust path between the source and the destination in VANETs. Existing routing protocols for MANETs (i.e. AODV[14], DSR [8], and GPSR[9]) are inadequate to achieve this goal. For example, AODV tries to build a minimum hop path from the source to the destination. It is not suitable in VANETs because each hop may be easily broken due to dynamic topology. One alternative approach is to always choose next hop that has the least bit error rate (BER) so that the link state of each hop is the most stable one. However, this approach may significantly increase the path length as well as the end-to-end delay. Thus it is desirable to *efficiently achieve robust packet delivery in VANETs*.

This paper proposed an efficient multi-hop routing protocol for urban area vehicular ad hoc networks, called Multi-hop Routing protocol for Urban vehicular ad hoc networks (MURU). MURU is a completely distributed protocol without the need of any pre-installed infrastructure. We use a novel metric called *expected disconnection degree* (EDD) to evaluate the quality of each candidate path between the source and the

¹For convenience, we use vehicle and node interchangeable in the rest of paper.

destination. The value of EDDs reflects the probability that the path would be broken in a certain time period, and is determined by the information of the predicted velocity and moving trajectory of each node along the path. The primary goal of the EDD design is to find the path that consists of the intermediate nodes that have stable relative position during a certain time period. As a result, despite the dynamic nature of VANETs, MURU can still achieve high packet delivery ratio and low packet delay. We evaluate the performance of MURU with extensive simulations. Comparing to existing routing protocols [14], [8], [9], MURU has much better performance in terms of packet delivery ratio, control overhead and packet delay.

The rest of this paper is organized as follows. In Section II, we give the related work. Section III describes the details of MURU. Section IV shows the performance evaluation results, and Section V concludes the paper.

II. RELATED WORK

In recent years various of ad hoc routing protocols have been proposed and can be classified into two categories: reactive (on-demand) and proactive protocols [2]. Because of less control overhead, reactive protocols such as AODV [14] and DSR [8] have gained more popularity. In AODV [14] each node maintains a routing table to keep the id of the next hop intermediate node. In DSR [8] the source determines the complete sequence of nodes to the destination based on the information of *Route Reply* message from the destination, and the route is listed in the header of each data packet transmitted. Both AODV and DSR try to find the shortest path in terms of the number of hops from the source to the destination. They both perform well in ad hoc networks without highly dynamic change of topology [2]. However, due to the characteristics of VANETs mentioned in Section I, they perform poorly in VANETs.

Location-based routing protocols have been proposed to make use of the geographical location information of nodes to support efficient and scalable routing in ad hoc networks. The geographic distance routing (GEDIR) [17] applies a greedy method to achieve efficient and loop-free data delivery in a collision free network. Karp and Kung [9] proposed GPSR that uses greedy perimeter forwarding to get around voids. Both GEDIR and GPSR work well in static ad hoc networks (e.g. sensor networks) but they did not consider the impact of high mobility.

In the past two years, VANET routing protocols are getting more attention. Chen *et. al.* [3] and Zhao *et. al.* [19] studied carry-and-forward schemes for sparse VANETs where disconnection happens frequently. However, they do not focus on routing in dense VANETs. In [13], mobile gateways are introduced by the integration of WLANs and 3G networks to connect vehicular nodes to the Internet. The PAVAN protocol proposed in [5] gives a mechanism to predict the available contents in a certain area and a two-tire service architecture is used so that control messages are transmitted in low-rate cellular networks and data packets are transmitted in ad

hoc networks. These works rely on infrastructure (i.e. 3G) networks. Different from existing works, MURU is a fully distributed location-based routing protocol that is tailored to VANETs by considering the characteristics of VANETs. It targets at efficiently utilizing wireless channels to provide robust data delivery with high throughput and low latency.

III. THE MURU PROTOCOL

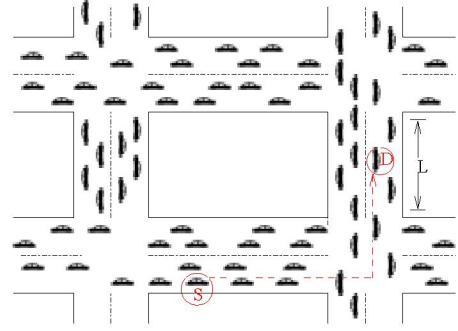


Fig. 1. The network topology

A. System Model

We consider a VANET in an urban area (e.g. the city of Miami). As shown in Figure 1, we model the area as a Manhattan style grid with a fixed block size across the area. All streets are assumed to be two-way. We assume that each vehicle's behavior is homogeneous. Although the direction and velocity of a vehicle on a certain time may depend on the layout of the road, traffic density and the individual behavior of driver, similar to [13], the mobility pattern of each vehicle is approximated by first order Markov chain shown in Figure 2.

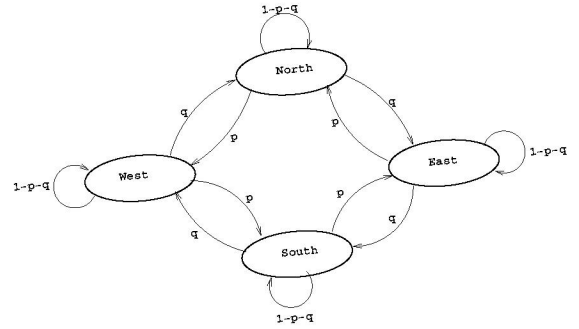


Fig. 2. The mobility pattern of nodes in vehicular ad hoc networks

The mobility information of each vehicle can be characterized by the average speed and the movement trajectory, which is determined by the destination and the road geometry. Vehicles communicate with each other through short range wireless channels (100m-250m) with WiFi network interfaces. Each vehicle is assumed to be equipped with global position system (GPS) and digital maps (e.g. MapMechanics [12]), and then a vehicle knows its location and street-level road

geometry at any instance. With advanced location registration and lookup services [11], [7], we assume that a source node is able to get the location of any destination nodes without incurring much traffic overhead.

B. Overview of MURU

Even though VANETs have unique characteristics such as the unstable channel condition and high mobility of each vehicle, we can still find some spaces to improve the performance of routing in VANETs. Specifically, since the mobility of each vehicle is usually constrained by roadmap geometry, the movement trajectory can be predicted according to the roadmap and the destination's location. With the information of movement trajectory and the average speed, we can predict the location of a group of vehicles, which can be the nodes on the path from the source to the destination, over a certain time period. Based on this observation, every a time period, a most robust path from the source to the destination can be found. Then the cumulated paths (in the time domain) provide the overall robust end-to-end data delivery in VANETs. We use the metric *expected disconnection degree* (EDD) to evaluate the probability that a path would be broken in a predefined time period. EDD is calculated according to the information of the speed and movement trajectory of each vehicle. Similar to AODV and DSR, MURU is an on-demand routing protocol. The source initiates the route discovery by sending *Route Request* message. Each candidate intermediate node performs receiver-based link quality estimation, and updates the current value of the path's EDD. The destination finally selects the path with the smallest EDD, which means the path has the smallest probability of path broken. By utilizing the roadmap geometry, we optimize the MURU with a backoff mechanism to reduce the control overhead by suppressing unnecessary control messages.

C. The Details of MURU

1) *Trajectory-constrained Route Request*: Suppose a source node wants to send data packets to a destination node, the shortest trajectory from the source to the destination can be calculated with the roadway geometry plus source and destination's location as well as mobility information. MURU tries to find a routing path using the shortest trajectory as the guidance. Due to the restriction of roadway geometry, the relative position of destination to source resides in a finite set. In case of grid layout, as shown in Figure 4, the relative position between the destination and the source must be within one of the following eight cases: *east, northeast, north, northwest, west, southwest, south or southeast*. When the relative position is east, north, west or south, the shortest trajectory maybe be a segment without turning points. Otherwise, the shortest trajectory would a segmented line with turning points.

The information of the shortest trajectory can be used to reduce control overhead of routing. In particular, a restricted area, which is called *broadcast area* is calculated based on the shortest trajectory to limit the broadcast range of routing request packets. Only nodes within the broadcast area may be

Notations:

n_i : Node i ;
 p_{req} : The route request packet;
 p_{rpl} : The route reply packet;
 p : The data packet;

Upon n_i receiving p_{req} from n_{i-1} :

```

if ( $n_i$  is out of the broadcast area defined in Eq. (1))
    Drop  $p_{req}$  and return;
if ( $n_{i-1}$  is closer to the destination)
    Drop  $p_{req}$  and return;
/*Assume  $n_s$  is the source*/
Calculate  $EDD(i-1, i)$  and  $EDD_{path}(s, i)$  with Eqs. (8)
and (9)
if  $\exists (n_j | j \neq i-1 \wedge EDD_{path}(s, j) < (EDD_{path}(s, i-1) + EDD(i-1, i)) \wedge n_j$  is closer to the destination than  $n_i$ )
    Drop  $p_{req}$  and return;
else if  $\exists (n_k | k \neq i-1 \wedge (EDD_{path}(s, k) + EDD(k, i)) < (EDD_{path}(s, i-1) + EDD(i-1, i)))$ 
    Drop  $p_{req}$  and return;
else
    Update the routing table and let link  $(n_{i-1}, n_i)$  join the path;
    Set the back-off timer with Eq.(10);
Upon  $n_i$  receiving  $p_{rpl}$ :
if ( $n_i \neq n_s$ )
    update routing table on  $n_i$ ;
    send out  $p_{rpl}$ ;
Upon  $n_i$  receiving  $p$ :
if (The next hop node is reachable)
    Forward  $p$  to the next hop node;
else
    Buffer  $p$  and perform local repair by sending out a new  $p_{req}$ 
    to the destination;

```

Fig. 3. The pseudocode of MURU

a possible intermediate node on the path. Each broadcast area should be a rectangle and the broadcast area can be easily calculated according to the location of the current sender, denoted by n , and the destination, denoted by dst . Formally, the broadcast area can be defined as:

$$\begin{aligned}
 Rectangle.X_{left} &= \min(n.X, dst.X) - L \\
 Rectangle.X_{right} &= \max(n.X, dst.X) + L \\
 Rectangle.Y_{top} &= \min(n.Y, dst.Y) - L \\
 Rectangle.Y_{bottom} &= \max(n.Y, dst.Y) + L
 \end{aligned} \tag{1}$$

Where L is the system parameter and is usually equal to the length of street blocks.

Once receiving the route request packet in which the trajectory information is piggybacked, the receiving node will process the packet if itself locates in the broadcast area. Otherwise, the node simply drops the packet.

The route request packet header fields MURU uses are shown in Figure 5. A MURU route request packet headers

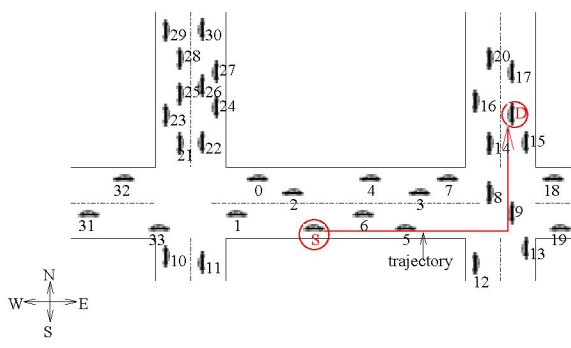


Fig. 4. The trajectory from node S to node D

Source ID	Destination ID	Sequence No.	LastHop ID
LastHop Location		LastHop Direc	
LastHop Velocity		EDD _{path}	
Shortest Trajectory			

Fig. 5. The format of a route request packet in MURU

include the mobility information (including velocity and direction) of the node who sends the packet most recently. Velocity and direction of a node may change from time to time and can be estimated by itself. The data structure of the shortest trajectory includes the coordinates of the start point, turning point and end point in the roadmap. A Route request packet header also carries the value of EDD_{path} , which is used to represent the quality of the route from the source to the current node. The definition of EDD_{path} will be described in Section III-C.4.

2) *Path Evaluation*: Traditional routing protocols (i.e. AODV) that tries to find the shortest path is not suitable in VANETs because of the highly dynamic network topology and channel condition. On the other extreme, we can select the path with the smallest bit error rate of each link. However, it may significantly increase the number of hops, which incurs a long end-to-end delay. Furthermore, due to the rapid change of network topology, the path breakage probability increases as the number of hops becomes large. We conduct a mathematical analysis to show that the robustness of a path is a concave function of the hop distance in VANETs. The analysis is as follows.

Considering the direct reflection path and ground reflection path between the two mobile nodes, two-ray ground reflection model is used to calculate the received power at distance d_{link} [16]:

$$P_r = \frac{P_t G_t G_r H_t^2 H_r^2}{d_{link}^4 l} \quad (2)$$

where $l(l \geq 1)$ is the system loss, h_r and h_t are the heights of the transmit and receive antennas respectively.

In reality, the received signal quality is determined not only by the received power P_r , but also by the thermal noise and inter-node interference (INI). Similar to [4], the average link SNR (signal-to-noise ratio) is used to evaluate the link quality with Rayleigh fading. The link SNR can be calculated as follows :

$$SNR_{link} = \frac{\alpha^2 P_r}{P_{thermal} + \alpha^2 P_{INI}} \quad (3)$$

where P_r is the received power at the end of a link, $P_{thermal}$ is the thermal noise power, P_{INI} is the sum of received power from all the interferers and α is the amplitude values of the fading channel with Rayleigh distribution.

Suppose the binary phase shift keying (BPSK) modulation is used, the bit error rate (BER) of the link with Rayleigh fading can be calculated as follows [4]:

$$BER_{link}^{Rayleigh} = \frac{1}{2} \left(1 - \sqrt{\frac{SNR_{link}}{1 + SNR_{link}}} \right) \quad (4)$$

In VANETs, The link quality between a pair of nodes may significantly decrease when either node turns at a corner, since the line-of-sight transmission will not be available. Considering the mobility of nodes in VANETs, we can calculate the packet error rate (PER) over a link as follows:

$$PER_{link} = 1 - (1 - BER_{link})^L + f(mo) \quad (5)$$

where L is the packet length and $f(mo)$ is the function of node mobility. With the Markov chain shown in Figure 2, the probability that the packet transmission fails is equal to the probability that either node leaves its current direction, which means, for each PER, $f(mo) = 2 * (p + q) - (p + q)^2$.

Considering the link-layer retransmission, the calculation of PER is adjusted as follows:

$$PER_{link}^N = 1 - \sum_{i=0}^N (1 - PER_{link}) PER_{link}^i \quad (6)$$

where N is the expected number of retransmission required per link.

Suppose that each hop has the same PER given in Equation (6) and that path has n hops, the PER of the path is calculated as follows:

$$PER_{path} = 1 - (1 - PER_{link}^N)^n \quad (7)$$

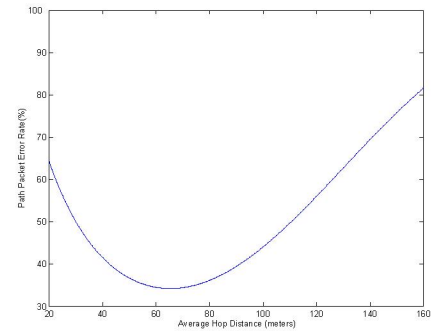


Fig. 6. The PER as a function of average hop distance

From Equations (2) (4) (5), and (6), we can see that PER_{path}^N is a function of the hop distance d_{link} . We give the numerical results of PER_{path}^N as the function of d_{link} in Figure 6. As shown in Figure 6, PER_{path}^N is a concave function so it has the minimal value, which corresponds to an *optimal hop distance*. From the figure, it can be seen that, to have the least path breakage probability, the hop distance should not be too small or too large.

3) *Calculating EDD*: With the analysis results in the previous section, we design a new metric called the *expected disconnection degree* (EDD) to predict the link breakage probability of each hop along the path over a certain time period T , which is a system parameter. EDD is calculated according to the predicted information of the speed, the movement trajectory, the location of each vehicle, as well as the roadway geometry. The path with a small aggregated EDD means the path has a good robustness over the time period T . Specifically, suppose the path consists of k nodes, denoted by $n_1 \dots n_k$. The EDD of link $n_{i-1}, n_i, 2 \leq i \leq k$, denoted by $EDD_{i-1,i}$, is calculated by n_i as follows:

$$EDD_{i-1,i} = \alpha * |D(i-1, i) - D_0|^l + \beta * f(L(i), T(1, k)) + \gamma * g(M(i-1), M(i), T(1, k))$$

where α, β and γ are the pre-specified tuning parameters. l is the path loss exponent and is determined by the propagation model used in urban areas. D_0 is a correction factor, which is a specified hop distance that produces a low bit error rate for the link. D_0 can be obtained by calculating the distance whose corresponding bit error rate is below a certain threshold under the propagation model. $D(i, j)$ is the geographic distance from n_i to n_j , $L(i)$ is the current location of node n_i , $M(i)$ is n_i 's predicted movement information including the expected direction and velocity during the time period T , and $T_{s,d}$ is the shortest trajectory from the source (n_s) to the destination (n_d). $f(L(i), T_{s,d})$ returns 0 if n_i is on the shortest trajectory towards the destination for a time period longer than T , and 1 otherwise. $g(L(i-1), M(i), T(1, k))$ returns 0 if n_{i-1} and n_i are expected to be within the transmission range of each other for a time period longer than T , and 1 otherwise. Some people may doubt why the functions f and g are discrete functions. The definitions of f and g have been extensively evaluated with simulations and we found that the discrete functions works better than continuous functions.

Suppose the source n_0 tries to deliver packets to destination and there is not an available path to the destination, the source initiates a route request packet and fills the appropriate information into all the fields in the packet header. The field $EDD_{path}(0, i)$ is used to evaluate the path quality from n_0 to n_i . $EDD_{path}(0, i)$ is calculated as follows:

$$EDD_{path}(0, i) = \begin{cases} 0, & i = 0 \\ EDD_{path}(0, i-1) + EDD(i-1, i), & \text{else} \end{cases} \quad (9)$$

Where the value of $EDD_{path}(0, i-1)$ is piggybacked in the EDD_{path} field in the route request packet header.

4) *Self-Pruning Data Requests*: In MURU, route request packets take a portion of the total control overhead. We design a self-pruning mechanism to reduce the control overhead of route discovery procedure without sacrificing the optimality of routes. In particular, when node n_i receives a route request p_{req} , it does not forwards the packet immediately. Instead, n_i holds the route request for a short time period (in milliseconds), which is denoted by $T_{backoff}^i$ and is defined as:

$$T_{back-off}^i = \sigma * EDD(i-1, i) \quad (10)$$

Where σ is the system parameter. Intuitively, if $EDD(i-1, i)$ is large, n_i would defer sending the route request message to exploit the possibility of finding a better path. During the backoff period, if n_i overhears a route request packet with the same tuple of $\langle SourceID, seqno \rangle$ from n_j , it further check if the $EDD_{path}(s, j)$ is smaller than $EDD_{path}(s, i)$ and n_j is closer to the destination or if the sum of $EDD_{path}(s, j)$ and $EDD(j, i)$ is smaller than $EDD_{path}(s, i)$. If so, n_i simply drop p_{req} and cancel the backoff timer. In this way, we can reduce control overhead since p_{req} on n_i becomes redundant. On the other hand, the node with a shorter backoff period will forward the request packet earlier than those with longer (8)backoff period, and then has a higher probability to be chosen as an intermediate node.

5) *Properties of MURU*: To show that the correctness of MURU, We give the following theorems.

Theorem 1: MURU is loop-free

Proof: We prove it by contradiction. Suppose there exists a loop in the path, and nodes n_a and n_b are in the circle, which means packets is forwarded from n_a to n_b and then is routed back to n_a . Suppose the source is n_s . Since the packet is forwarded from n_a to n_b , according the the routing algorithm of MURU, the $EDD_{path}(s, a)$ is smaller than $EDD_{path}(s, b)$. Since the packet is then forwarded from n_b back to n_a , we can see that $EDD_{path}(s, b)$ is smaller than $EDD_{path}(s, a)$, which leads to contradiction. ■

Theorem 2: MURU always finds the path from source S to destination D with smallest EDD.

Proof: We prove it by contradiction. Suppose path A has the smallest EDD_{path} from the source to the destination and MURU selects another path B with a larger EDD. From Eq. (10), it is easy to see that the aggregated backoff time of a path is proportional to the value of EDD_{path} .

Consider the first case that path A and B do not have any joint intermediate nodes. Since the total back-off time of B is larger than the total back-off time of A (Note that path B has a larger EDD_{path}), the route request packet that sent along path A arrives at the destination earlier than the route request travels along path B. Therefore, MURU should select A, which leads to contradiction.

In the second case that path A and path B have joint intermediate nodes. Suppose node n_i is the first joint intermediate node of path A and path B. Thus, path B can be divided into

two sub-paths: the sub-path from the source to n_i , and the sub-path from n_i to the destination, denoted by path B_1 and B_2 respectively. Similarly, path A can be divided into two sub-paths: the sub-path from the source to n_i , and the sub-path from n_i to the destination, denoted by path A_1 and A_2 respectively. Since path A has smaller EDD_{path} than path B, path A_1 must have a smaller EDD than path B_1 . Otherwise, path B_1 and path A_2 compose the path from S to D with the smallest EDD_{path} , which contradicts to the assumption that path A has the smallest EDD_{path} . From the result obtained in the first case, the route request packet sent through path B_1 arrives at n_i later than that sent through path A_1 . Since each node only maintains single path from source to itself, only path A_1 will be kept in n_i , which means that path B will not be selected.

Theorem 3: The route repair mechanism initiated by node i finds the routing path with the smallest EDD_{path} from the source to the destination among all paths that have node i as an intermediate node.

Proof: The routing path from the source to the destination can be divided into two sub-paths, the path from the source to node i and the path from node i to the destination, denoted by path A and B respectively. To prove that the routing path from the source to the destination has the smallest EDD_{path} is equivalent to prove that path A has the smallest EDD_{path} among all paths from the source to node i and path B has the smallest EDD_{path} among all paths from node i to the destination.

From Theorem 2, we have proved that MURU always finds path with the smallest EDD_{path} from the source to the destination. The route repair mechanism of MURU tries to build a routing path from node i to the destination, therefore path B has the smallest EDD_{path} among all routing paths from node i to the destination.

Suppose there is another path C that has a smaller EDD_{path} from the source to node i than path A, then path A would not be selected by MURU as the sub-path of the path from the source to the destination because MURU always finds the path with smallest EDD_{path} . This is contradict with our assumption that path A has been chosen as the sub-part of the path from the source to the destination.

IV. PERFORMANCE EVALUATION

In this section, we present the simulation results to show that MURU outperforms other routing protocols in terms of higher packet delivery ratio, lower control overhead and lower packet delay. The impact of the parameters used in Equation 8 is also examined.

A. Simulation Setup

We build the simulation using ns2 [15] simulator. The area is assumed to be on a $700m \times 700m$ square area presenting a grid layout, where the street layout is in grids with block length

of $100m$. A number of vehicles are deployed to the streets and the initial location of each vehicle is randomly chosen to reflect the even distribution on the map. The mobility of vehicles follows the mobility pattern shown in Figure 2 with average speed of 45 miles per hour. IEEE 802.11 DCF protocol is used as the MAC layer transmission protocol. We compare MURU with DSR [8], AODV [14] and GPSR [9]. Instead of the standard AODV, we actually use AODV-LL [2] which eliminates the periodical HELLO messages, and then saves control overhead. Even though this version of AODV makes the link breakage detection on-demand, [2] shows that AODV-LL performs significantly better than the standard AODV. The performance metrics are: packet delivery ratio, total control overhead and end-to-end packet delay. Performance of each protocol is evaluated in two different scenarios, in which the locations of the source and the destination are set to be fixed and mobile respectively. The total number of nodes in the region varies from 60 to 150. Each source is assumed to initiate a constant bit rate (CBR) flow that sends data packets periodically. The packet size is 512 bytes. The simulation time is 100 seconds and each case is repeated 40 times to achieve a high confidence of the results.

B. Simulation Results and Analysis

1) *Scenario 1: Fixed Source and Destination:* In the first scenario, a fixed source sends packets to a fixed destination located four blocks away from source. Figure 7 (a) shows the data delivery ratio as the function of the total number of nodes. As shown in the figure, the data delivery ratios of AODV, DSR and GPSR are lower than 50%, which proves that they are not suitable for urban vehicular ad hoc networks. Such low ratios are caused by the highly dynamic link quality due to the mobility of each node. Furthermore, the data delivery ratios of AODV, DSR and GPSR keep the same or slightly increase as the network density increases, which means AODV, DSR and GPSR cannot take the advantage of network density to alleviate the impact of mobility to the link quality. The reason behind is that AODV, DSR and GPSR only focus on finding the shortest path from the source to the destination. In contrast, the data delivery ratio of MURU is much greater than 50% as the number of mobile nodes is greater than 80 and the ratio significantly increases as the network density increases. This is because that high network density provides MURU more opportunities to find a path whose links are quite robust.

Figure 7 (b) shows the overhead of routing protocols measured in the total number of routing packets sent. The overhead of GPSR is proportional to the number of nodes because each node periodically sends beacons in GPSR. Since no beacon is sent in AODV, DSR and MURU, the overhead of these three reactive protocols are much lower than that of GPSR. Since MURU has a larger data delivery ratio, the average overhead needed per packet of MURU is much lower than that of DSR and AODV.

Figure 7 (c) shows the average end-to-end delay. Since GPSR is a stateless routing protocol, which does not have route discovery, the delay of GPSR is the smallest between

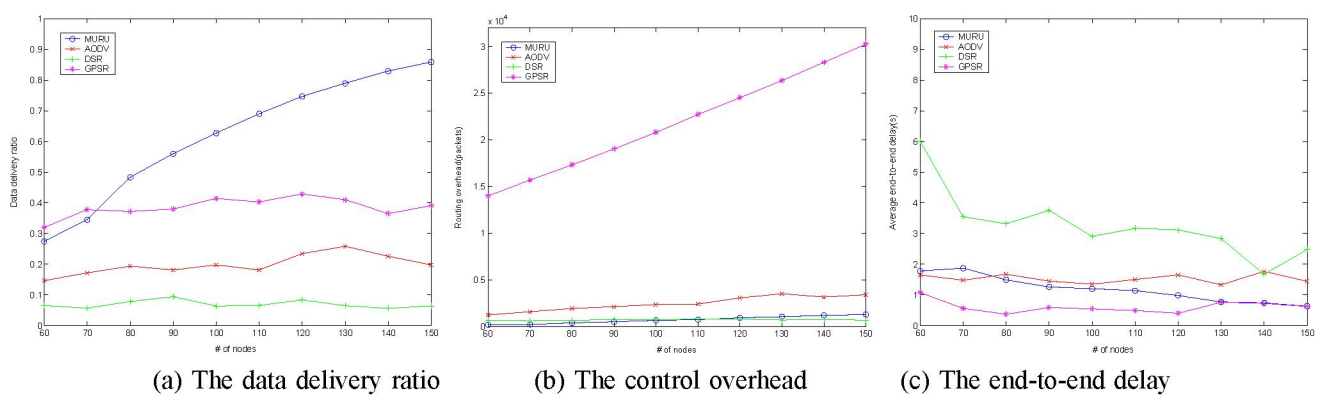


Fig. 7. The performance comparison in scenario 1

the protocols we evaluated. The delay of MURU decreases as the network density increases. It is because that a higher network density leads to a more robust path. When the number of nodes in the network reaches 130, MURU performs as well as GPSR. Both DSR and AODV have the higher average delay than MURU by the fact that paths built in DSR and AODV are vulnerable to the change of network topology.

2) *Scenario 2: Mobile sources and destinations:* In this scenario, three mobile source-destination pairs are selected. The sources and destinations are randomly chosen and the mobility pattern of them follows Figure 2. Figure 8 (a) shows the data delivery ratio. As can be seen, MURU outperforms other three protocols significantly. Its delivery ratio is larger than 90% when the number of nodes in the simulation is more than 110. The data delivery ratio of GPSR decreases as the increase of network density because the higher network density provides more opportunities for a node to select next hop closer to destination, which degrades the link quality due to link distances. The performances of DSR and AODV are poor in this scenario and the increase of network density doesn't bring any benefit to the data delivery ratio. This is mainly because of the highly dynamic network topology.

Figure 8 (b) shows the average routing overhead. Similar to Scenario 1, the overhead of GPSR is proportional to the number of nodes moving in the network. The overhead of other three routing protocols are much lower than that of GPSR since no beacons are sent. Basically, the overhead of DSR keeps the same as the network density increases because the overhead of DSR is mainly determined by the data delivered during the simulation which is basically keeps the same too. The overhead of AODV increases slightly as the increase of network density by the fact that more nodes participates in the finding of routing paths as the network density increases. The increases of data delivered introduces the increases of the total overhead needed in MURU. However, the use of trajectory constrained and self-tuning route request greatly reduce the overhead of MURU significantly. Since MURU has a larger data delivery ratio, it is obvious that MURU outperforms DSR, AODV and GPSR.

Figure 8 (c) shows the average end-to-end delay. Compared

with Figure 7 (c), we can see that DSR has a much lower delay when both source and destination are mobile. The reason behind is that, in Scenario 1, DSR performs poorly since the highly dynamic network topology poisons freshness of the cached paths. In Scenario 2, since the network topology (especially the relative locations of the source and the destination) changes more quickly, it is more difficult for DSR to build multiple-hop path and most of the paths built by DSR are one hop. Therefore the average end-to-end delay in DSR is very low. Similarly, the delay of GPSR follows the same trend mentioned in Scenario 1. MURU has lower average delay than that of AODV and GPSR. The delay of MURU decreases as the network density increases and the reason has been explained in the previous section.

C. Choosing Parameters

We have set parameters α , β and γ used in Equation 8 to be 0.075, 0.1 and 0.1 respectively. These three parameters represent the weight of different factors that have different impact on the the quality of each link. In this section, we evaluate the significance of each parameter in the performance of data delivery ratio. Due to the limitation of space, we only show part of our evaluation in which α is fixed at 0.075 and different values are set to β and γ under Scenario 1.

Figure 9 (a) shows the data delivery ratio with the value of α and γ are set to be 0.075 and 0.1 respectively. As β is changed from 0.015 to 0.15, the data delivery ratio doesn't shows many significant changes. Figure 9 (b) shows the data delivery ratio with the value of α and β are set to be 0.075 and 0.1 respectively. We can see that the data delivery ratio is not sensitive to the value of γ . From these evaluation, we find that the performance of MURU is not sensitive to the values of system parameters in Equation 8.

V. CONCLUSION

In this paper we proposed a routing protocol, called MURU, for urban area vehicular ad hoc networks. MURU is designed to find robust paths to delivery data with high data delivery ratio, low control overhead and low packet delay. A new metric *expected disconnection degree* (EDD) is introduced to predict the probability that a link would be broken in a certain time

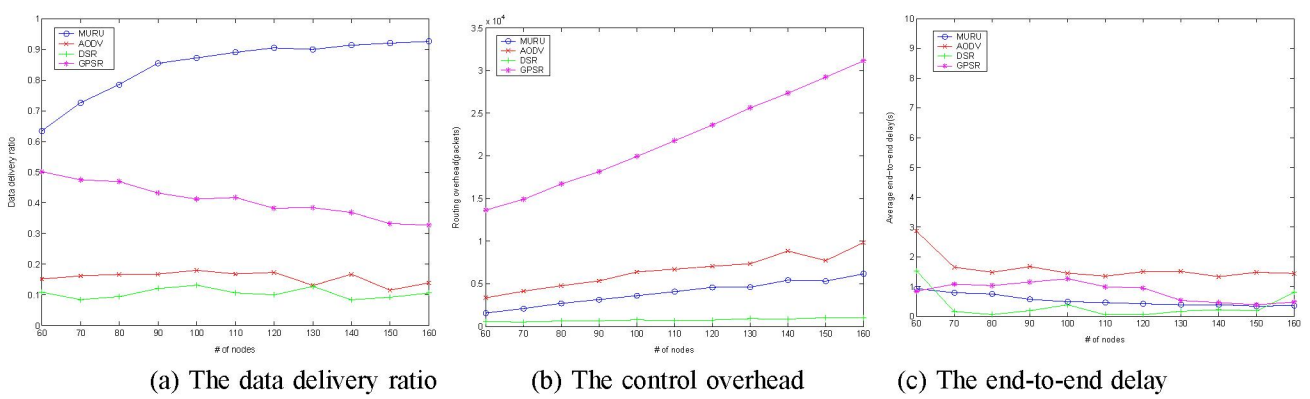


Fig. 8. The performance comparison in scenario 2

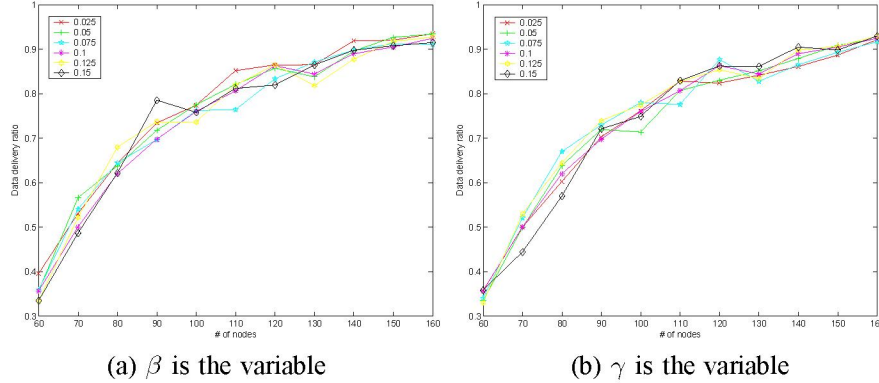


Fig. 9. The data delivery ratio under different values of β and γ

period based on the mobility information of vehicles. With EDD, MURU finds robust paths for end-to-end data delivery. MURU is a fully distributed protocol that does not require any pre-installed infrastructure. Comparing to most popular ad hoc routing protocols, our simulation results demonstrate that the MURU has much better performance in terms of packet delivery ratio, data packet delay and algorithm overhead.

REFERENCES

- [1] M. Bechler, W.J. Franz, and L. Wolf, "Mobile Internet Access in FleetNet," *Kommunikation in Verteilten Systemen*, 2003.
- [2] J. Broch, D.A. Maltz, D.B. Johnson, Y-C Hu, and J. Jetcheva, "A Performance Comparison of multi-hop wireless ad hoc network routing protocols," in *ACM MOBICOM'98*, Oct. 1998.
- [3] Z. Chen, H. Kung, and D. Vlah, "Ad Hoc Relay Wireless Networks over Moving Vehicles on Highways (short paper)," in *ACM MobiHoc'01*, 2001.
- [4] G. Ferrari and O.K. Tonguz, "Impact of Clustering on the BER Performance in Ad Hoc Wireless Networks," in *International Zurich Seminar on Communications (IZS): Access-Transmission-Networking*, 2004.
- [5] S. Ghandeharizadeh, S. Kapadia, and B. Krishnamachari, "PAVAN: A policy framework for content availability in vehicular ad-hoc networks," in *ACM Workshop on Vehicular Ad Hoc Networks (VANET)*, Oct. 2004.
- [6] D. K. Goldenberg, J. Lin, A. S. Morse, B. E. Rosen, and Y. R. Yang, "Towards mobility as a network control primitive," May 2004, pp. 163–174.
- [7] M. Grossglauser and M. Vetterli, "Locating nodes with ease: Last encounter routing in ad hoc networks through mobility diffusion," in *IEEE INFOCOM'03*, Oct. 2003.
- [8] D. Johnson and D. Maltz, "Dynamic source routing in ad hoc wireless networks," *Mobile Computing*, 1996.
- [9] B. Karp and H. T. Kung, "GPSR: Greedy perimeter stateless routing for wireless networks," in *ACM MOBICOM'00*, Aug. 2000.
- [10] K. Kutzner, J.-J. Tchouto, M. Bechler, L. Wolf, B. Bochow, and T. Luckenbach, "Connecting Vehicle Scatternets by Internet-Connected Gateways," *Multiradio Multimedia Communications*, 2003.
- [11] J. Li, J. Jannotti, D. Decouto, D. Karger, and R. Morris, "A Scalable Location Service for Geographic Ad-hoc Routing," in *ACM MobiCom'00*, 2000.
- [12] MapMechanics, "GB Traffic Volumes," www.mapmechanics.com, May 2005.
- [13] V. Nambodiri, M. Agarwal, and L. Gao, "A study on the feasibility of mobile gateways for vehicular ad-hoc networks," Oct. 2004.
- [14] C.E. Perkins and E.M. Royer, "Ad hoc on demand distance vector (AODV) routing," in *2nd IEEE Workshop on Mobile Computing Systems and Applications*, Feb. 1999.
- [15] Vint Project, "http://www.isi.edu/nsnam/ns/."
- [16] T.S. Rappaport, *Wireless Communications, Principles and Practice*, Prentice Hall, 1996.
- [17] I. Stojmenovic and X. Lin, "GEDIR: Loop-Free Location Based Routing in Wireless Networks," in *International Conference on Parallel and Distributed Computing and Systems*, 1999.
- [18] X. Yang, J. Liu, F. Zhao, and N. Vaidya, "A vehicle-to-vehicle communication protocol for cooperative collision warning," in *Int'l Conf. on Mobile and Ubiquitous Systems: Networking and Services (MobiQuitous 2004)*, Aug. 2004.
- [19] J. Zhao and G. Cao, "VADD: Vehicle-Assisted Data Delivery in Vehicular Ad Hoc Networks," in *IEEE INFOCOM'06*, 2006.