

ACAR: Adaptive Connectivity Aware Routing for Vehicular Ad Hoc Networks in City Scenarios

Qing Yang · Alvin Lim · Shuang Li ·
Jian Fang · Prathima Agrawal

Published online: 22 May 2009
© Springer Science + Business Media, LLC 2009

Abstract Multi-hop vehicle-to-vehicle communication is useful for supporting many vehicular applications that provide drivers with safety and convenience. Developing multi-hop communication in vehicular ad hoc networks (VANET) is a challenging problem due to the rapidly changing topology and frequent network disconnections, which cause failure or inefficiency in traditional ad hoc routing protocols. We propose an adaptive connectivity aware routing (ACAR) protocol that addresses these problems by adaptively selecting an optimal route with the best network transmission quality based on statistical and real-time density data that are gathered through an on-the-fly density collection process. The protocol consists of two parts: 1) select an optimal route, consisting of road segments, with the best estimated transmission quality, and 2) in each road segment of the chosen route, select the most efficient multi-hop path that will improve the delivery

ratio and throughput. The optimal route is selected using our transmission quality model that takes into account vehicle densities and traffic light periods to estimate the probability of network connectivity and data delivery ratio for transmitting packets. Our simulation results show that the proposed ACAR protocol outperforms existing VANET routing protocols in terms of data delivery ratio, throughput and data packet delay. Since the proposed model is not constrained by network densities, the ACAR protocol is suitable for both daytime and nighttime city VANET scenarios.

Keywords connectivity aware routing · vehicular ad hoc networks · connectivity model · trajectory based routing

1 Introduction

Wireless communication among moving vehicles is increasingly the focus of research in both of the academic research community and automobile industry, driven by the vision that exchange of information among vehicles can be exploited to improve the safety and comfort of drivers and passengers [2, 9, 20, 29]. Some automobile manufacturers have equipped their new vehicles with global positioning systems (GPS), digital maps and even wireless interfaces, e.g. Honda-ASV3. In addition, the federal communications commission (FCC) has allocated 75 MHz of spectrum in the 5.9 GHz band for vehicle-vehicle and vehicle-roadside communication, called dedicated short range communications (DSRC). IEEE is also working on the IEEE

Q. Yang (✉) · A. Lim · S. Li · J. Fang
Computer Science and Software Engineering,
Auburn University, Auburn, AL 36849, USA
e-mail: yangqin@auburn.edu

A. Lim
e-mail: limalvi@auburn.edu

S. Li
e-mail: lishuan@auburn.edu

J. Fang
e-mail: fangjia@auburn.edu

P. Agrawal
Electrical and Computer Engineering, Auburn University,
Auburn, AL 36849, USA
e-mail: pagrawal@eng.auburn.edu

1609 family of standards for wireless access in vehicular environments (WAVE), which define an architecture and a complementary, standardized set of services and interfaces that collectively enable secure vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) wireless communications. Although IEEE 1609.3 considers the networking layer and provides an alternative for IPv6, it does not define the ad hoc routing protocol between vehicles, and has left this issue open.

Though several technical problems need to be solved before installing vehicular networks, in the near future, large scale vehicular networks will be available to provide people with more conveniences in their driving experience. For example, through such networks, people can query the price and services provided by gas stations in a certain region, or remotely control their smart-houses [15] while driving home. Drivers can even download a real-time traffic image from a traffic camera located at a certain point, or connect to access points of parking lots to inquire the number of available parking slots. These types of applications could tolerate some delay, e.g. a few minutes. If the information could be successfully retrieved from the remote server, it would be very helpful and desirable to drivers.

To realize this vision, we must first select the most appropriate architecture. There are three broad categories of network architectures: infrastructure-based, ad hoc networks and hybrid. The infrastructure-based architecture takes advantage of the roadside infrastructure or existing cellular networks. However, a big issue of such networking is the high operation cost. Moreover, the cellular networks have other drawbacks such as the limited bandwidth and symmetric channel allocation for uplink and downlink. Ad hoc networks do not need infrastructure, so the cost of building such network will be very low and it can even operate in the event of disasters. The hybrid architecture is more practical which combines these two architectures by considering vehicles as data relays between roadside base-stations [7, 40]. This architecture also requires the function of multi-hop communication between vehicles, which is the essential part of ad hoc network architecture. This paper focuses on the vehicular ad hoc network (VANET) architecture with the flexible deployment and self-organizing capabilities.

Due to special characteristics of VANETs, traditional routing protocols in wireless ad hoc networks may not be suitable for vehicular communications. For example, DSR [17] and AODV [28] are not suitable for VANETs because of the large route maintenance overhead. Therefore, some variants of stateless geographic routing protocols, such as [18, 24], may be the best

choices. However, even with geographic routing, many of the following challenges still need to be addressed:

1. Dynamic and rapidly changing topologies of vehicular networks can cause frequent communication disconnections among vehicles. As revealed in [36], the frequent network disconnection is the most important issue in designing protocols for VANET.
2. Geographic forwarding protocols select the shortest route (minimal number of hops) that may suffer from a higher packet error rate due to the poor link quality of each hop.
3. The uneven distribution of vehicles on the roads makes route selection more complex, e.g. the shortest path in terms of geographic distance may experience more frequent network disconnections.
4. Some protocols [38, 41] make use of the density information on roads to select routes but the inaccuracy of statistical data may cause routes to be incorrectly computed.
5. Because of obstacles to wireless signal by large objects, e.g. skyscrapers in cities, communication between vehicles must have line-of-sight.

To address these problems, we propose a new routing protocol called adaptive connectivity aware routing (ACAR). There are four main contributions in this paper. First, based on the statistical information on the road (e.g. number of vehicles and average velocity), we proposed a connectivity model that provides the probability of network connectivity on a road segment. This connectivity model also takes into account the phenomena that (red) traffic lights can block approaching vehicles and those nodes will move as a platoon in the next road segment. Second, we introduced a transmission quality model that combines the network connectivity probability and data delivery ratio of packets being forwarded along a road segment. Third, as the statistical data may not be accurate, an on-the-fly information collection algorithm is developed to help ACAR adaptively select the best route. Fourth, instead of using greedy geographic forwarding, we proposed a scheme for selecting the optimal next hop that ensures the highest end-to-end data delivery ratio.

The remainder of this paper is organized as follows. Section 2 discusses currently available routing protocols for VANET. Then in Section 3, we describe the assumptions and system model for ACAR. In Sections 4 and 5, the routing strategy and simulation results are presented, respectively. Section 6 gives the conclusions.

2 Related work

There exist several routing protocols that can be applied to vehicular ad hoc networks as summarized in [3, 16, 22]. They can be grouped into two categories: 1) those that assume the networks are always connected and 2) those that focus on intermittently connected networks.

Protocols in the first category are suitable for the urban rush hour scenarios, where vehicles are densely packed and locating a node for forwarding a message is typically not an issue. However, traditional ad hoc routing protocols (e.g., AODV [28] and DSR [17]) have poor route convergences and low communication throughputs because they are adversely affected by the highly dynamic nature of node mobility as shown by the results in [27].

Since a GPS device will be a standard component in future vehicles, more position-based routing protocols have been proposed for VANETs [5, 18, 23–26, 37]. Position-based approaches use geographic coordinates information or relative positions of nodes to generate an efficient route through the network. For example, the greedy perimeter stateless routing (GPSR) [18] protocol may be a good choice because it is stateless and performs well despite high mobility in VANETs. However, GPSR may encounter the problems of selecting incorrect next hops due to out-of-date neighbors information, routing loop and too many (detour) hops as stated in [24]. In [24], packets are forwarded along the *Dijkstra* shortest path as calculated from road maps. Similarly, in MDDV [37], the forwarding trajectory of a message is determined as the trajectory that minimizes the sum of weights on that graph between the source and a vertex in the destination region. Moreover, the authors [5] developed protocols that disseminate information to a set of target zones, rather than specific destination nodes. They utilize a propagation function whose value is minimized over the target zones. Unlike other greedy position-based unicast routing protocols, anchor-based street and traffic aware routing (ASTAR) [23] utilizes city bus routes as a strategy to find routes with a high probability for delivery.

All the above protocols omit the problem of network disconnection. The authors in [25] introduced a new metric, expected disconnection degree (EDD), to evaluate the probability that a candidate route would be broken. By broadcasting the RREQ message, the path with the smallest EDD will be selected as the route. To handle the problem of mobile end nodes (source or sink), [26] adapts the idea of guards which automatically adjust the connectivity path when end

nodes change their speeds and/or directions. However, it first needs to broadcast the route discovery request to the entire network to find a proper route, causing excessive networking overhead even with some optimization schemes. In summary, all these approaches basically require networks to be fully connected; otherwise, the route discovery phase will fail, rendering the subsequent routing strategy useless. Nevertheless, this assumption is often not true in VANET, as it was concluded in [36] that network partitions in VANET are very frequent.

Assuming networks are not always connected, another group of routing protocols are proposed in the literature [1, 7, 19, 21, 32, 38, 41]. These routing protocols can be considered as the delay tolerant protocols and the carry-and-forward [4] scheme is used when network disconnection happens. Network disconnections occur frequently in rural highway situations and in cities at night where fewer vehicles are running, making establishing end-to-end routes impossible. Even in densely-populated urban scenarios, sparse sub-networks can also be prevalent.

To route a message from a vehicle to a roadside unit, the motion vector (MOVE) routing algorithm [19] uses knowledge of neighboring vehicles velocities and trajectories to predict which vehicle will physically travel closest to the fixed destination. Another knowledge-based scheme, scalable knowledge-based routing (SKVR) algorithm [1] utilizes the relatively predictable nature of public transport routes and schedules. The SKVR works in two levels: the top level is inter-domain routing, where a source and destination are on different bus routes, while the bottom level consists of intra-domain routing within the same bus route.

When network infrastructures are available at intersections, a static node assisted adaptive routing protocol (SADV) has been proposed [7] for vehicular networks. When disconnected, each static node has the capability to store a message until it can forward the message to a node traveling on the optimal path. Optimal paths are determined based on a graph abstracted from a static road map and weighted with expected path forwarding delays from a delay matrix.

Similar to other routing algorithms designed for delay-tolerant networks, the geographical opportunistic routing protocol (GeOpps) [21] uses the navigation information to route packets efficiently. GeOpps assumes that each vehicle has a navigation system that provides a suggested route to a traveling destination. Each neighbor vehicle will use a utility function built into the navigation system to calculate the amount of

time required to reach the next interest point. The vehicle that can deliver the packet fastest or closest to the destination will be chosen as the next hop for the message. Those protocols either require infrastructure at intersections or vehicles following the navigation system, but these assumptions may not be true in reality.

Assuming a pure vehicular ad hoc network architecture, the VADD [41] protocol is proposed. When wireless connectivity is not available, the carry-and-forward strategy is used to transfer packets along vehicles on the fastest roads available. Since vehicles may deviate from predicted paths, the routing path should be recomputed continuously during the forwarding process. To aid in this process, VADD uses a street graph weighted with expected packet delivery delays. However, a drawback is that when the average distance between vehicles is close to the communication range, the transmission delay will be much longer than the expected one used in VADD. Unlike VADD, a delay-bounded routing protocol [32] is introduced for VANET. The goal of this routing algorithm is to select an optimal path that not only has the least transmission cost but also meets the delay requirement given by the application. However, the delay model used in [32] still has a similar problem as VADD.

Existing VANET routing protocols omit the connectivity information in highly dynamic networks, though mobility can increase the capacity of ad hoc wireless networks [12]. Obviously, mobility is the distinguishing feature of vehicular networks, affecting the evolution of network connectivity over space and time in a unique way. The mathematical connectivity model in ad hoc networks has been studied in [8, 30] with the assumption that nodes follow the poisson distribution. However, node movement in VANET can be affected by multiple factors such as the traffic lights, vehicles moving around and speed limits. So instead of using those traditional mobility models (e.g. the Random Waypoint model), researchers proposed several mobility models for VANETs [11, 31, 34, 35]. In the constant speed motion (CSM) model [34], a generic vehicle i 's movement is constrained on a given road topology, and its speed is set to $v_i = v_{min} + (v_{max} - v_{min})\alpha$ where α is a uniformly distributed random variable in $[0, 1]$. The fluid traffic motion (FTM) model [31] adopts a traffic stream approach on a microscopic level. It describes the speed as a monotonically decreasing function of vehicular density, forcing a lower bound on speed when the traffic congestion reaches a critical state. Then based on the intelligent driver model (IDM) [35], IDM with intersection management (IDM-IM) and IDM with lane changing (IDM-LC) models were proposed

in [11]. The IDM-IM is a flows-interaction model which adds intersection handling to the car-to-car interaction description provided by IDM; the IDM-LC further extends the flows-interaction description of IDM-IM, by adding overtaking capability to vehicles. To the best of our knowledge, the IDM-based mobility models are the most accurate ones for VANET. A detailed analysis of those IDM-based models is described in [10], and a simulator, VanetMobiSim [14], based on these models is developed by the authors.

Although there exist some efforts to create accurate mobility models, such as the IDM with lane changing model [11], most of these models are too complicated to be used in the networking protocol design. Instead of microscopic mobility models, we look at VANET in a macroscopic way and try to reveal the statistical property of network connectivity. In the design of the ACAR protocol, this information is used to select the route with the highest probability of connection and thus the network throughput is increased.

3 Problem statement and system model

3.1 Problem statement

The topology of VANET has a unique characteristic—it consists of one or more sub-graphs (one sub-graph if the network is fully connected) of the road map topology. Previous researches in wireless ad hoc networks often make an unrealistic assumption on nodes mobility. For example, with the most popular Random Waypoint model, nodes can freely move within a certain area with randomly chosen velocities. However, nodes in VANET do not have the ability to roam freely without regards to obstacles and traffic regulations, i.e. all road segments containing vehicles construct the VANET topology. Therefore, the problem of efficient routing of packets in VANETs can be transformed into selecting a route with the highest throughput from the road map.

Consider the network situation shown in Fig. 1, where the source node at the bottom left corner is trying to send packets to the destination at the top right corner. In this figure, the lengths of road segment $I_A I_B I_C$, $I_A I_C$, $I_A I_D$ and $I_C I_D$ are 1200 m, 1000 m, 707 m and 707 m, respectively. The numbers of nodes deployed on each above-mentioned road segment are 22, 9, 5 and 2, respectively. All vehicles move with the average velocity of 10 m/s.

With the GPSR protocol, packets will be forwarded through a multi-hop route (part of such a route is depicted as dashed lines with arrows). Because the

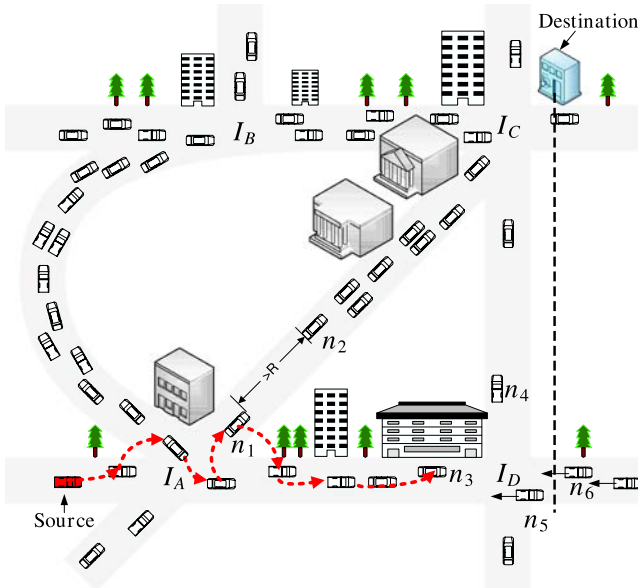


Fig. 1 Illustration of the routing problem in VANETs

network density on road segment $I_A I_C$ is low, disconnections will happen frequently. For example, node n_1 in Fig. 1 fails to communicate with node n_2 as they are out of communication range. In this case, GPSR enters the perimeter mode and selects nodes on road segment $I_A I_D$ to forward packets. However, since network partitions are very common in VANETs, GPSR may face network disconnections again. For instance, because wireless signal may be blocked by objects, e.g. skyscraper in the city, the communication between node n_3 and n_4 may be impossible due to the absence of line-of-sight. That implies the GPSR protocol may take many detours to find connected route, e.g. on segment $I_A I_B I_C$, after many perimeter mode searches. If there is no such connected route in networks, GPSR may search through the entire networks and finally fail to find a route.

To make use of road map information, the geographic source routing (GSR) protocol [24] was proposed for VANETs. With this approach, road segment $I_A I_C$ will be selected to forward the packets. Because the assumption of connected networks does not always hold, the GSR may fail to deliver packets when network partitions occur. If the carry-and-forward scheme [4] is added into GSR, packets can finally reach the destination. However, the delay of forwarding packets on this road segment will be higher than routing packets along $I_A I_B I_C$. According to measurements in our simulations, the network connectivity probabilities of road $I_A I_C$ and $I_A I_B I_C$ are .29 and .84, respectively. The .29 connectivity probability can be interpreted as the network is disconnected 71% of the time, so the net-

work delay can be simply calculated as $.71 \times (1000/v) + .29 \times (1000/c)$ where v is the average velocity of vehicles on road $I_A I_C$ and c is the wireless transmission speed. As $v \ll c$, the delay of forwarding packets along $I_A I_C$ is $delay_{AC} \approx (710/v)$. Similarly, the delay of forwarding packets on $I_A I_B I_C$ is $.16 \times (1200/v) + .84 \times (1200/c) \approx (192/v)$. Therefore, routing packets along $I_A I_B I_C$ generates a much smaller delay than that of $I_A I_C$.

In the motion vector (MOVE) [19] protocol, the packet carrier will select the next hop that is currently or will be closest to the destination; otherwise, it will carry (buffer) the packet until a next hop is available. It provides 7 rules for current node to select the next hop, and one of them states that if the current packet carrier is in AWAY state and one neighbor is in TOWARDS state, packets must be forwarded to this neighbor. For instance, as shown in Fig. 1, node n_5 (moving away from the destination) will forward packets to n_6 as it move toward the destination. However, if n_6 moves over the vertical dashed line, it enters the AWAY state and will forward packets back to following vehicles that are in the TOWARDS state. This situation is so-called Ping-Pong effect, and it will be solved if no more following vehicles become available. However, this problem becomes worse when the network density is higher.

To select a route with the minimal transmission delay, [41] proposes the vehicle-assisted data delivery (VADD) protocol for VANETs. According to the protocol, since the network density on road $I_A I_C$ is equal to $1/R$, the delay of forwarding packets on $I_A I_C$ is $d_{AC} = \alpha \cdot l_{AC}$ where $l_{AC} = 1000$ m is the length of road $I_A I_C$ and α is a constant. Similarly, $d_{AB} = \alpha \cdot 1000$, so we have $d_{AB} = d_{AC}$. As stated in VADD, if the packet carrier (vehicle) at intersection I_A chooses to deliver packets on road $I_A I_B$, the expected packet delivery delay from the intersection I_A to the destination is:

$$D_{AB} = \frac{1}{1 - P_{AB} \cdot P_{BA}} \cdot (d_{AB} + P_{BA} \cdot d_{BA} + P_{BA} \cdot P_{AC} \cdot d_{AC} + P_{BC} \cdot d_{BC}) \quad (1)$$

where P_{AB} is the probability that the packet is forwarded through $I_A I_B$ at intersection I_A , which is smaller than 1. Since $P_{AB} \cdot P_{BA} < 1$, we have $D_{AB} > (d_{AB} + P_{BA} \cdot d_{BA} + P_{BA} \cdot P_{AC} \cdot d_{AC} + P_{BC} \cdot d_{BC})$, so $D_{AB} > d_{AB}$. On the other hand, since $D_{AC} = d_{AC} = d_{AB}$, we obtain $D_{AB} > D_{AC}$. Therefore, road $I_A I_C$ will be chosen by VADD to forward packets as it has the smallest expected delivery delay. However, the delay of sending packets along road $I_A I_B I_C$ is actually the lowest.

Therefore, to select the optimal route in VANETs, a proper model of the network connectivity is very important and it is determined by several factors such as network density, road length and number of lanes on roads. In this paper, we first model the network connectivity and then propose an approach to select the optimal route that can achieve the highest network throughput.

3.2 Assumptions

As GPS and navigation systems are becoming standard equipment in vehicles, we assume every vehicle can obtain its current location. We also assume vehicles are installed with a pre-loaded digital map, such as the commercial map provided by MapMechanics, which not only describes the land attributes such as road topology and traffic light period but also is accompanied by traffic statistics such as traffic density and average velocity at a certain time of the day. These digital maps with statistical data are derived from billions of GPS sampled points from vehicles on the move. Similar digital maps can also be found from the Internet, e.g. yahoo.com. We expect more accurate and detailed digital maps to be invented and equipped on vehicles in the future. We also assume the vehicles are of similar sizes and each vehicle is equipped with a 802.11 wireless interface.

3.3 Connectivity model of road segment

The connectivity model of road segment, the portion of a street between two adjacent intersections, is investigated in this section. We first propose the cell-based connectivity model for vehicles moving within road segments and the cluster-based connectivity model for vehicles clustered around intersections. Then, we integrate those two models and present the connectivity model for a road segment.

3.3.1 Cell-based connectivity model

We first consider the model for the one-lane case and later generalize it to multiple lanes. In the one-lane scenario, we divide the road segment equally into m cells so that each cell can contain at most one vehicle and each vehicle can occupy only one cell. The length of cell d can be set as the average length of vehicles, e.g. 5 m. It will be fairly common that a vehicle partially occupies two adjacent cells. In this case, the cell containing the majority part of this vehicle is considered occupied. Since the distance between occupied cells will be used to compute the distance between vehicles in these cells, we found that there would be an error (at most 5 m)

in the distance computation. However, compared to the large wireless communication range, e.g. 250 m in 802.11b and 1000 m in DSRC, this error can be ignored. Therefore, the probability of connectivity of networks can be formulated as follows:

If there are n vehicles (also called nodes) on a road segment, what is the probability that the distance of any two neighboring nodes is less than the communication range $R = n_0 \cdot d$, i.e. there are no more than n_0 successive empty cells on the road.

In one-lane scenarios, the number of empty cells is always $m - n$; but in the case of multiple lanes, the number of empty cells will range from $m - n$ to $m - \lceil n/n' \rceil$ where n' is the number of lanes. For multiple lanes, each cell in the road may contain any number of nodes within $[0, n']$. So in the extreme case, if every occupied cell contains only one node, the number of empty cells is $m - n$. On the other hand, if each occupied cell has n' nodes, the number will become $m - \lceil n/n' \rceil$. For instance, suppose 5 vehicles are deployed into a road with 5 cells and 3 lanes. Let cells be ordered geographically such that cell c_0 is at the leftmost and c_4 is at the rightmost position. It may happen that 3 vehicles are located in cell c_0 and the other two in cell c_4 . So the number of empty cells in this case is 3. Intuitively, if the number of empty cells k is equal or less than n_0 , then the network must be connected. If $k > n_0$, the network may be connected or disconnected depending on how the empty cells are distributed.

We denote P_{dis} and $P_{\text{con}} = 1 - P_{\text{dis}}$ as the probability of network being disconnected and connected, respectively. Since it is not easy to compute P_{con} , we first calculate P_{dis} . To obtain this probability, two other probabilities are required: 1) empty cell probability $P1$ that there exist exactly k empty cells if n nodes are deployed into m cells, denoted as $P1 = P\{\mu(n, m) = k\}$, and 2) successive empty cell probability $P2$ that there exist more than n_0 successive empty cells given exactly k empty cells on the road segment, which is denoted as $P2 = P\{\varphi(m, k) > n_0\}$. Then the probability that the network is disconnected becomes:

$$P_{\text{dis}} = \sum_{k=\max(m-n, n_0)}^{\max(m-\lceil n/n' \rceil, n_0)} P\{\mu(n, m) = k\} \cdot P\{\varphi(m, k) > n_0\} \tag{2}$$

Empty cell probability P1. To drive safely on roads (with one lane), a driver need to keep a certain distance from the front or rear vehicles, thus the occupancy of one cell is dependent on the adjacent cells. Considering multiple lane cases, since traffic flows on different

lanes are independent of each other, the dependency of occupied cells is broken. If there are two moving directions on roads, the occupied cells will be more randomly distributed. Therefore, we first assume that vehicles are uniformly deployed on roads, then adjust our model taking into account the clustering (platoon) phenomena of vehicles.

With the assumption of uniformly distributed nodes, we investigate the probability that there exist exactly k empty cells on the road. Suppose there are n nodes deployed on the road with m cells. Let A_i be the event that the i th cell is empty, and let \bar{A}_i be the event complementary to A_i (i th cell is occupied). Then we have:

$$P\{\mu(n, m) = k\} = \sum_{1 \leq i_1 < \dots < i_k \leq m} P\{A_{i_1} \dots A_{i_k} \bar{A}_{j_1} \dots \bar{A}_{j_{m-k}}\} \tag{3}$$

where $\{j_1, j_2, \dots, j_{m-k}\} = \{1, 2, \dots, m\} - \{i_1, i_2, \dots, i_k\}$. $P\{A_{i_1} \dots A_{i_k} \bar{A}_{j_1} \dots \bar{A}_{j_{m-k}}\}$ is the probability that the i_1 th to i_k th cells are empty and the j_1 th to j_{m-k} th cells are occupied by nodes. Since every term on the right side of the above equation is the same, the total number of them is C_m^k . Moreover, we can rewrite the term as:

$$P\{A_{i_1} \dots A_{i_k} \bar{A}_{j_1} \dots \bar{A}_{j_{m-k}}\} = P\{A_{i_1} \dots A_{i_k}\} \cdot P\{\bar{A}_{j_1} \dots \bar{A}_{j_{m-k}} | A_{i_1} \dots A_{i_k}\} \tag{4}$$

where $P\{A_{i_1} \dots A_{i_k}\} = \frac{C_{m-k}^{n'}}{C_m^n}$ is the probability that there exist at least k empty cells on this road, and $P\{\bar{A}_{j_1} \dots \bar{A}_{j_{m-k}} | A_{i_1} \dots A_{i_k}\}$ is actually the probability of $P\{\mu(n, m - k) = 0\}$. So we can obtain the following recursive formula:

$$P\{\mu(n, m) = k\} = C_m^k \cdot \frac{C_{(m-k) \cdot n'}^n}{C_{m \cdot n'}^n} \cdot P\{\mu(n, m - k) = 0\} \tag{5}$$

Notice that the probability that there exists at least one empty cell is:

$$P\{\mu(n, m) > 0\} = P\left(\bigcup_{i=1}^m A_i\right) = \sum_i P(A_i) - \sum_{i < j} P(A_i A_j) + \sum_{i < j < h} P(A_i A_j A_h) - \dots \tag{6}$$

So the probability that all cells are occupied is:

$$P\{\mu(n, m) = 0\} = \sum_{l=0}^m C_m^l \cdot (-1)^l \frac{C_{(m-l) \cdot n'}^n}{C_{m \cdot n'}^n} \tag{7}$$

By substituting Eq. 7 into 5, the probability that there exist exactly k empty cells can be computed.

Successive empty cell probability P2. The $P\{\varphi(m, k) > n_0\}$ denotes the probability that there exist more than n_0 successive empty cells on the road given that there are exactly k empty cells. Since the number of occupied cells is $m - k$, we are able to formulate this problem as:

Consider throwing k items into $N = m - k + 1$ bags and each bag can contain any number of items $0, 1, \dots, k$, then what is the probability that at least one bag contains at least $(n_0 + 1)$ items.

Since it is hard to directly compute this probability, we first examine the case where all bags satisfy the condition:

C1: Every bag contains at most n_0 items.

We denote $\text{Num}(k, N)$ as the number of possible deployments that satisfy **C1**. Then it can be rewritten as:

$$\begin{aligned} \text{Num}(k, N) &= \text{Num}(k, N - 1) + \text{Num}(k - 1, N - 1) \\ &\quad + \text{Num}(k - 2, N - 1) + \dots \\ &\quad + \text{Num}(n_0, N - 1) \end{aligned} \tag{8}$$

The proof of Eq. 8 is stated as follows. Let us consider a certain bag, b_i , that may contain $0, 1, \dots, n_0$ items. Suppose it contains j items, then the number of deployments that satisfy **C1** is $\text{Num}(k - j, N - 1)$. By summing up all the possible j , we obtain:

$$\text{Num}(k, N) = \sum_{j=0}^{k-n_0} \text{Num}(k - j, N - 1) \tag{9}$$

Since each term in the right part of Eq. 9 can be expanded as

$$\text{Num}(k - j, N - 1) = \sum_{l=0}^{k-j-n_0} \text{Num}(k - j - l, N - 2) \tag{10}$$

After expanding each term, Eq. 9 becomes:

$$c[0]^{N-1} \cdot \text{Num}(k, 1) + c[1]^{N-1} \cdot \text{Num}(k - 1, 1) + \dots + c[k]^{N-1} \cdot \text{Num}(0, 1) \tag{11}$$

where $\text{Num}(x, 1)$ refers to the number of possible deployments of putting x items into one bag.

$$\text{Num}(x, 1) = \begin{cases} 0, & x > n_0 \text{ or } x < \max\{0, k - n_0 \cdot (N - 1)\} \\ 1, & \max\{0, k - n_0 \cdot (N - 1)\} \leq x \leq n_0 \end{cases} \tag{12}$$

This number will be 0 if $x > n_0$ or $x < k - n_0 \cdot (N - 1)$, since **C1** does not hold in these cases. If $x < 0$, it means putting negative number of items into bags, so $\text{Num}(x, 1)$ is also 0; otherwise, $\text{Num}(x, 1) = 1$.

Then the number of deployments satisfying **C1** will be the sum of coefficients of all terms whose value are 1, i.e.

$$\sum_{i=k-n_0}^{\min\{k, (N-1) \cdot n_0\}} c[i]^{N-1} \tag{13}$$

$$c[i]^{t+1} = \sum_{j=\max\{0, i-n_0\}}^{\min\{i, t \cdot n_0\}} c[j]^t \tag{14}$$

where $c[i]^1 = 1$ ($i = 0, 1, \dots, n_0$). Since the total number of all possible deployments is $C_{N+k-1}^k = C_m^k$, the probability $P2$ is:

$$P\{\varphi(m, k) > n_0\} = 1 - \frac{\sum_{i=k-n_0}^{\min\{k, (m-k) \cdot n_0\}} c[i]^{m-k}}{C_m^k} \tag{15}$$

Substituting Eqs. 5 and 15 into 2, we can calculate the probability of the network being disconnected or connected on a certain road, provided the network density information is known.

3.3.2 Cluster-based connectivity model

Since traffic lights (red signals) block approaching vehicles, these vehicles will form a cluster (or convoy) on the road. Therefore, the proposed connectivity model that assumes uniform node distribution needs to be modified by adjusting the network density information.

As shown in Fig. 2, suppose on road segment A, there are n_A nodes moving toward the intersection. Assume the length of A is l_A , the average velocity of vehicles on it is v_A and time period of red traffic light

is t_A . Then the expected number of vehicles stopped by every red light on road A is:

$$\overline{m}_A = \begin{cases} \frac{n_A \cdot v_A \cdot t_A}{l_A}, & (v_A \cdot t_A) < l_A \\ n_A, & \text{otherwise} \end{cases} \tag{16}$$

If $(v_A \cdot t_A) \geq l_A$, the red signal t_A is long enough so that all vehicles on A are blocked. When the light turns green, stopped vehicles will resume moving. Those moving in the same direction will be very close to each other because drivers prefer to follow the traffic flow. As a result, we can assume those vehicles move as a cluster. As the distance between vehicles in a cluster is short, the network composed of these nodes is considered connected. Therefore, the number of nodes on roads needs to be modified because clustered nodes will be regarded as one node.

Since nodes in one cluster cannot fit into one cell, they are spread over several cells. For example, assume there are \bar{n} nodes in a cluster and they uniformly distribute on each lane of a road. Then, the total number of cells on this road is reduced from m to $m - \lceil \bar{n}/n' \rceil \cdot (d_s/d)$, where d_s is the safeguarded distance between vehicles, d is the length of a cell and n' is the number of lanes. If nodes are uniformly deployed on each lane, $\lceil \bar{n}/n' \rceil$ will be the maximum number of nodes on each lane, and $\lceil \bar{n}/n' \rceil \cdot (d_s/d)$ the maximum number of cells occupied by this cluster. The safeguarded distance between vehicles can be calculated by:

$$d_s = v \cdot t_r + v^2/(2b) + d \tag{17}$$

where v is the average velocity, t_r is the reaction time and b is the deceleration value of comfortable braking. Usually the distance between vehicles is larger than the safeguarded distance for safe driving, so we use the upper bound value $\lceil \bar{n}/n' \rceil$ to denote the number of cell occupied by these clustered vehicles.

Next, we investigate how to compute the number of nodes, \bar{n} , in each cluster. Suppose the numbers of nodes moving toward the intersection on road segment A, B, C and D are n_A, n_B, n_C and n_D , respectively. Then for each vehicle on A, the probability of moving to D will be:

$$p_{AD} = \frac{n_D}{n_B + n_C + n_D} \tag{18}$$

Suppose there are m_A nodes blocked on road A, the expected number of these blocked nodes moving from A to D is:

$$\overline{m}_{AD} = \frac{m_A \cdot n_D}{n_B + n_C + n_D} \tag{19}$$

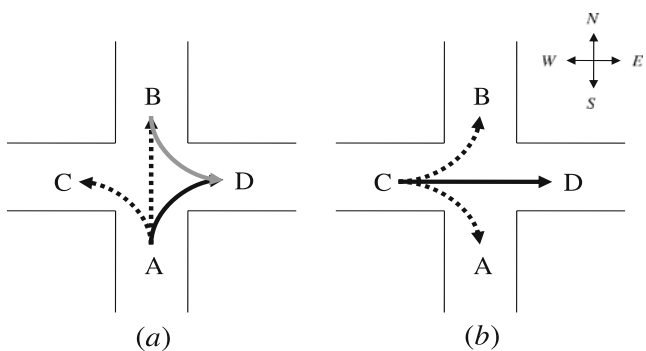


Fig. 2 Illustration of how traffic lights affecting the connectivity model (a, b)

In the same way, we can get $\overline{m_{BD}}$ and $\overline{m_{CD}}$. If the traffic light controlling south–north traffic turns green, as shown in Fig. 2a, $\overline{m_{AD}} + \overline{m_{BD}}$ nodes will move as a cluster on road D . If the traffic light controlling west–east traffic turns green, as shown in Fig. 2b, there will be a cluster of $\overline{m_{CD}}$ nodes moving on road D . Therefore, the number of nodes on road D is reduced to:

$$n_D - (\overline{m_{AD}} + \overline{m_{BD}} + \overline{m_{CD}} + 2) \tag{20}$$

During each traffic light period, two clusters will be produced. Therefore, the number of clusters on road D is:

$$N_D = \begin{cases} \left\lceil \frac{2 \cdot l_D}{v_D \cdot T} \right\rceil, & l_D > (T \cdot v_D) \\ 1, & \text{otherwise} \end{cases} \tag{21}$$

where T is the traffic light period at this intersection. When $l_D > (T \cdot v_D)$, new clusters are generated before the first cluster moves out of road D . So the number of clusters is $\left\lceil \frac{2 \cdot l_D}{v_D \cdot T} \right\rceil$, which is the upper bound of this N_D . With the newly computed number of node and cells, we can compute the probability of connectivity of road segment.

If two moving directions exist on road D , we need to modify the density on the opposite direction too. By adjusting the number of clusters, the cluster connectivity model is also suitable for one-way roads or roads with traffic light at only one end.

3.3.3 Integration of cell and cluster-based connectivity models

We have proposed the cell-based connectivity model where nodes move on roads without clustering and the cluster-based connectivity model in which traffic lights block vehicles to form clusters around intersections. Now, we describe how to integrate those two models to compute the connectivity of road segment.

Vehicles form a cluster when they are blocked by the traffic light in an intersection. However, the cluster will exist only for a period of time. After that, these vehicles will merge into the traffic flow of roads they are moving on. In other words, vehicles deployment on a road segment changes periodically between cluster-based and cell-based modes.

Suppose there is only one cluster on a road segment, e.g. the road segment A as shown in Fig. 2. Nodes in this cluster are geographically labeled as $1, 2, \dots, \bar{n}$, where node 1 is the closest one to the intersection and \bar{n} is the furthest one. Therefore, the size of this cluster is \bar{n} . Assume these nodes will move into another road, and the density and velocity of this road are d and

\bar{v} , respectively. We define t_i^b as the time for a node i ($i \in [1, \bar{n}]$) to move out of the cluster, i.e. after t_i^b seconds, node i will merge into the traffic flow of a road segment (e.g. D in Fig. 2).

To compute the time t_i^b of node i , we first investigate the one-lane one-cluster case, and then generalize it to multiple-lane multiple-cluster cases. Within one lane, a vehicle cannot accelerate freely as its movement is restricted by many factors: the distance to the preceding vehicle, velocities of the preceding vehicle and itself. This phenomena is represented by the car following model [35], in which the acceleration rate of node i at time instance t is:

$$a_i^t = \frac{dv_i^t}{dt} = a \left[1 - \left(\frac{v_i^t}{v^0} \right)^4 - \left(\frac{s_i^*}{s_i^t} \right)^2 \right] \tag{22}$$

where v_i^t is the velocity of node i at time t , a is the maximum acceleration rate and s_i^t is the distance between node i and its preceding node. v^0 is the desired speed, which is equal to \bar{v} in this case. Distance s_i^* is called *desired dynamical distance* [35] and is computed by:

$$s_i^* = s_0 + \left(v_i^t \tau + \frac{v_i^t \cdot \Delta v_i^t}{2\sqrt{ab}} \right) \tag{23}$$

It is a function of the minimum bumper-to-bumper distance s_0 , the minimum safe time headway τ , the velocity difference with respect to front vehicle $\Delta v_i^t = (v_i^t - v_{i-1}^t)$ and the maximum acceleration and deceleration values a and b . For node 1 in the cluster, its distance to the preceding node is $s_1 = 1/d$; because in the cell-based model, vehicles are assumed to be evenly distributed on road segments. The distance node i drives from time 0 to t is $l_i^t = \int_0^t \frac{1}{2} \cdot a_i^t \cdot t^2 dt$, so the value of s_i^t will be $(l_{i-1}^t - l_i^t)$.

Therefore, we can obtain the time t_i^b that node i needs to reach the speed of \bar{v} . It is computed by solving the integral equation:

$$\int_{t=0}^{t=t_i^b} a_i^t \cdot t dt = \bar{v} \tag{24}$$

During time period $[t_{i-1}^b, t_i^b]$, there are only $(\bar{n} - i + 1)$ nodes remaining in the cluster. According to Section 3.3.2, we can compute the new number of cells, so does the connectivity probability during time period of $[t_{i-1}^b, t_i^b]$. Then, the overall connectivity probability of the road segment can be computed as:

$$P_{cell} \cdot \frac{T - \max\{t_i^b\}}{T} + \sum_{i=1}^{i=\bar{n}} P_{cluster}(\bar{n} - i + 1) \cdot \frac{t_i^b - t_{i-1}^b}{T} \tag{25}$$

where $t_0^b = 0$, $T = l/\bar{v}$ is the time a vehicle needs to move from one end to the other end of the road segment. P_{cell} is the probability of connectivity computed by the cell-based model, and $P_{cluster}(\bar{n} - i + 1)$ is the probability of connectivity obtained through the cluster-based model with a cluster of $(\bar{n} - i + 1)$ nodes. If there are N_c clusters and the size of each cluster is \bar{n}_j , $j \in [1, N_c]$, the connectivity probability of road segment is:

$$\sum_{j=1}^{j=N_c} P_{cell} \cdot \frac{T - \max\{t_i^j\}}{T} + \sum_{j=1}^{j=N_c} \sum_{i=1}^{i=\bar{n}_j} P_{cluster}(\bar{n}_j - i + 1) \cdot \frac{t_i^j - t_{i-1}^j}{T} \tag{26}$$

where t_i^j is the time t_i^b that node i needs to move out of the j th cluster.

In multiple lane cases, we assume clustered vehicles are evenly distributed on each lane because it is natural for drivers to change lanes if the current one is too congested. We apply the calculation of the single lane case to each lane and can compute the value of t_i^j for every $i \in [1, \bar{n}_j]$ and $j \in [1, N_c]$. Note that, the value of each t_i^j will change, and so does $(t_i^j - t_{i-1}^j)$. However, with Eq. 26, we can compute the probability of connectivity of road segment for multiple lane and multiple cluster cases.

3.4 Connectivity model of route

So far, we modeled the connectivity probability of a certain road segment given the information on segment length, number of vehicles, average velocity and traffic light periods. However, since connectivity probabilities of two adjacent road segments are not independent, the connectivity probability of a route consisting of multiple road segments cannot be calculated as the product of connectivity probabilities of these segments. For example, assume there is a route consisting of n road segments and the connectivity probability of each segment is P_i ($i = 1, 2, \dots, n$). Then the connectivity probability of this route P_{rt} is not $\prod P_i$, the actual lower bound if P_i is not independent.

Suppose the length and average velocity of each road segment on this route are l_i and v_i , respectively; then the expected delay of forwarding packets along the path is:

$$\sum_{i=1}^n \left[\frac{P_i \cdot l_i}{c} + \frac{(1 - P_i) \cdot l_i}{v_i} \right] \tag{27}$$

where c is a constant denoting the speed of wireless transmission and $c \gg v_i$. This equation sums up the delay of forwarding packets on each road segment i ,

and gives the end-to-end delay of forwarding packets on the route. Similarly, if we consider the whole route as a segment, this delay can be calculated as:

$$\frac{P_{rt} \cdot l_{rt}}{c} + \frac{(1 - P_{rt}) \cdot l_{rt}}{v_{rt}} \tag{28}$$

where $l_{rt} = \sum l_i$ is the total length of this route and $v_{rt} = \sum v_i/n$ is the average velocity of vehicles moving on this route. Thus, the connectivity probability of this route P_{rt} is:

$$\frac{c}{c - v_{rt}} - \frac{c \cdot v_{rt} \sum_{i=1}^n \left[\frac{P_i l_i}{c} + \frac{(1 - P_i) l_i}{v_i} \right]}{(c - v_{rt}) l_{rt}} \tag{29}$$

As $c \gg v_{rt}$, the first part of the above equation should be very close to 1. Since the second part must be larger than 0, the value of P_{rt} must be smaller than 1. The value of P_{rt} might be smaller than 0 though it is a very unlikely case; if so, we consider $P_{rt} = 0$. In the above equation, we notice that the connectivity probability of a route is a decreasing function of the end-to-end delivery delay. That means if the average velocity and route length are the same, the protocol forwarding packets on a route with a higher P_{rt} will generate a lower data delivery delay.

3.5 Estimation of transmission quality

The carry-and-forward scheme can solve the problem of packets being dropped when network disconnections occur in VANETs. However, there are other factors that cause packets to be dropped in the delivery processes, such as number of hops, interference from other vehicles, and transmission collisions. Considering these factors, we propose a model for estimating data delivery ratio, Q_{rt} , for forwarding packets on a route. First, we describe the packet error rate (PER) model for a single hop. Then we model the packet error rate of delivering packets on a road segment. Finally, we describe the PER estimation of forwarding packets on a route consisting of several segments.

According to our connectivity model, for a certain road segment, the larger the network density, the higher is the probability of connectivity. However, higher densities can cause larger interferences (more nodes in interference range), and thus reduce the packet delivery ratio. Therefore, it is non-trivial to integrate the packet delivery ratio and connectivity probability to select the best route.

We first investigate the PER of single hop communication between two nodes. To model the path loss between those two nodes, two cases have to be considered: the line-of-sight (no other nodes between these

two) and non-line-of-sight (at least one neighbor between them) [39]. Because of the popularity and lower price of IEEE 802.11 devices, the physical layer in VANET (the DSRC/IEEE 802.11p PHY) will be a variation of the OFDM (orthogonal frequency-division multiplexing) based the IEEE 802.11a standard. So the channel fading model of determining the received signal power level in the case of line-of-sight (LOS) is:

$$P_r = \frac{P_t}{(4\pi)^2 \left(\frac{d}{\lambda}\right)^\gamma} \left[1 + \eta^2 + 2\eta \cos\left(\frac{4\pi h^2}{d\lambda}\right) \right] \quad (30)$$

where P_t is the transmit power, d is the distance between the transmitter and receiver, λ is the wavelength of propagating signal, η is the reflection coefficient of the ground surface, γ is the path loss factor and h is the antenna height. The model of non-line-of-sight (NLOS) is expressed as:

$$P_r = \begin{cases} P_t G_t G_r \left(\frac{\lambda}{4\pi}\right)^2 & (d \leq 1m) \\ P_t G_t G_r \left(\frac{\lambda}{4\pi}\right)^2 \cdot \frac{1}{d^\gamma} & (d > 1m) \end{cases} \quad (31)$$

Taking into account the effect introduced by the cyclical prefix attached to each OFDM symbol, the signal to interference plus noise ratio (SINR) should be reduced by a factor of α :

$$\text{SINR} = \alpha \cdot 10 \log_{10} \left(\frac{P_r}{P_0 + \sum_{i=1}^{N_{\text{INT}}} P_{\text{INT}}^i} \right) \quad (32)$$

where α is 0.8 according to [39], P_0 is the background noise, and P_{INT}^i is the interference from neighbor n_i .

Suppose on a certain road segment, as shown in Fig. 3, node n_a is sending packets to n_b and the distance between them is d_{ab} . Then from the perspective of n_b , there will be $\text{den} \cdot (R_{\text{INT}} - 2R - d_{ab})$ potential interfering nodes around it. In which, R and R_{INT} are the communication and interference ranges of n_b , and den is the network density of this road.

In the IEEE 802.11 protocols, before each communication the RTS/CTS (request to send/clear to send) packets need to be transmitted between sender and receiver to reduce frame collisions introduced by the hidden terminal problem. After that, during the communication between n_a and n_b , nodes within their communication ranges are not allowed to transmit packets. Thus, the potential interfering nodes must be in the area that is outside the communication ranges of n_a and n_b but inside their interference ranges. Within these areas, for a circle with a radius of R , there is at most

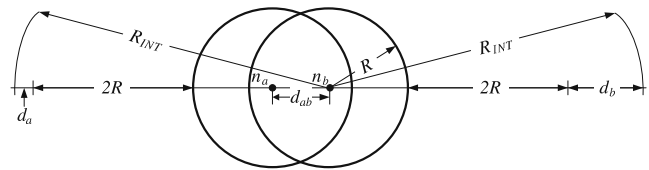


Fig. 3 Illustration of the number of potential interfering nodes

one transmission that can interfere with the packet receptions at n_b . Therefore, there are at most $\lceil \frac{R_{\text{INT}} - R}{2R} \rceil + \lceil \frac{R_{\text{INT}} - R - d_{ab}}{2R} \rceil$ transmissions that interfere with node n_b simultaneously.

The receiving power P_{INT}^i of each interference transmission can be computed through Eq. 30 or 31 where d is the distance between n_b and the center of each segment labeled as $2R$ in Fig. 3. For cases of $d_a < 2R$ and $d_b < 2R$, $3R + d_b/2$ and $3R + d_{ab} + d_a/2$ are the distances of interference transmissions in d_b and d_a , respectively. If node n_b is in a nearby intersection area, there will be more potential interfering nodes. Similarly, for roads with different network densities joined at an intersection, we can calculate the number N_{INT} .

In Eq. 32, we use the maximum number of interfering transmissions with the communication between n_a and n_b , thus the worst case of SINR for n_b is obtained. In simulations, we find this lower bound value is very close to the real one; thus, we use it to further calculate the bit error rate and packet error rate of a single hop transmission.

Suppose the binary phase shift keying (BPSK) scheme is used to modulate the signal, the bit error rate (BER) is:

$$\text{BER} = Q\left(\sqrt{2 \cdot \text{SINR}}\right) \quad (33)$$

where $Q(x) = 0.5 - 0.5 \times \text{erf}\left(\frac{x}{\sqrt{2}}\right)$ and $\text{erf}(\cdot)$ is the error function. Because of retransmissions in the link layer, the frame error rate (FER) can be computed as:

$$\text{FER}_{\text{link}} = 1 - \sum_{i=0}^N (1 - \text{FER}) \text{FER}^i \quad (34)$$

where $\text{FER} = 1 - (1 - \text{BER})^L$, L is the length in bits of each frame and N is the number of retransmission times. Suppose every packet is composed of t frames, the PER is computed by:

$$\text{PER} = 1 - (1 - \text{FER}_{\text{link}})^t \quad (35)$$

So given the communication distance and number of neighbors, we can model the PER of a single hop.

Next, we discuss how to model the PER of a certain road segment (denoted as PER_{rs}). On a certain

road segment, suppose there is a route $route_j$ that is composed of h hops with PER at every hop of PER_l ($l = 1, 2, \dots, h$), then the PER of forwarding packets along this route $route_j$ can be computed as:

$$PER_{route_j} = 1 - \prod_{l=1}^h (1 - PER_l) \tag{36}$$

This equation is valid only if the PER is independent from one hop to the next; but due to the wireless communication environment there could be interference which violates this assumption. However, in this paper, we use this equation as the first-order approximation of the PER of forwarding packets on a certain route. Since different routes (composed of different hops) give different PERs, we consider a routing algorithm that minimizes PER, so the problem is to determine the minimal expected PER. If there are n nodes and k' empty cells on the road, for a certain distribution of these empty cells, the minimal PER of this road segment is denoted as $PER_{k'}^i = \min\{PER_{route_j}\}$. This value can be easily determined because we can compute the PER of every route. Therefore, the expected value of PER_{rs} can be calculated as:

$$E[PER_{k'}^i] = E_k[E[PER_{k'}^i | k' = k]] \tag{37}$$

which can further be rewritten as:

$$PER_{rs} = \sum_{k=m-n}^{m-\lceil n/n' \rceil} \sum_{i=1}^{C_m^k} \frac{1}{C_m^k} \cdot PER_k^i \cdot P\{\mu(n, m) = k\} \tag{38}$$

where m and n' are the number of cells and number of lanes on this road segment, respectively. Thus we use $D_{rs} = 1 - PER_{rs}$ to model the data delivery ratio of a certain road segment. For a given route consisting of n road segments, suppose the data delivery ratio of every road segment is D_i ($i = 1, 2, \dots, n$), then the delivery ratio of this route D_{rt} can be computed as $\prod D_i$.

Therefore, transmission quality of a route is modeled as $Q_{rt} = D_{rt} \times P_{rt}$ where P_{rt} is the probability of network connectivity of a certain route. As it will be shown in Section 5, since the ACAR protocol chooses routes with the highest transmission qualities, the data delivery ratio and network throughput are drastically increased compared to other protocols.

4 Routing algorithm

The ACAR protocol includes two essential elements: 1) correctly selecting an optimal route consisting of

road segments with the best estimated transmission quality, and 2) efficiently forwarding packets hop-by-hop through each road segment in the selected route. To eliminate the impact of inaccurate statistical density data, we developed an adaptive route selection algorithm that collects real-time density information on-the-fly while forwarding packets. In each road segment in the selected route, the next hop is selected using a metric that minimizes the packet error rate (PER) of the entire route based on measured PERs at each node. In addition, carry-and-forward [4] mechanism is adopted to handle frequent network partitions in VANETs.

4.1 Neighbors location prediction

Through GPS, each vehicle can obtain its real-time location and velocity. This information is then broadcasted periodically along with its id to neighbors. Each node maintains a table of its neighbors information including locations, velocities and ids. To avoid out-of-date neighbors, we implemented the neighbor location prediction (NLP) algorithm proposed in [33], each node predicts its neighbors positions using following formulas:

$$\begin{aligned} x' &= x + ratio \cdot (x - x_o) \\ y' &= y + ratio \cdot (y - y_o) \end{aligned} \tag{39}$$

Where $ratio = (t_b - t + t_o)/t_b$, t is current time, t_b is beacon period and t_o is the time when previous beacon message was received from the same neighbor. (x, y) , (x_o, y_o) and (x', y') are the neighbor's current, previous and predicted position, respectively. Then, only those still within the communication range are considered for next hop selections.

4.2 Adaptive route selection

If the density information on each road segment is correct, the optimal route will be the one with the highest transmission quality. However, in reality, there may be some errors in the statistical density data. For example, suppose on road A there are 100 nodes (on average) in the afternoon, then it is possible that the network density between 2:00 pm–4:00 pm is 50 and from 4:00 pm to 6:00 pm is 150.

One possible solution to this problem is to flood the entire network to collect the real-time density information. However, even with directional and efficient flooding, this approach could still cause too much broadcast overhead. Therefore, we propose an adaptive path selection approach that collects real-time

density data when packets are being forwarded into the network.

ACAR first computes a route based on statistical density data from the pre-loaded map. It then puts the route information into packet headers and transmits packets along this selected route. While the packets are being forwarded to the destination, network densities of all road segments along this path are collected simultaneously. This process, called on-the-fly density collection, is described in the next section. After a pre-defined number of on-the-fly density collections (e.g. 10), the density information on road segments in the route can be obtained at the destination. If the error rates of some road segments density exceed the threshold, e.g. 30%, the sink node sends an acknowledge message to notify the source about the updated density of that road. Next, the source node re-computes a new route based on the recently received and more accurate density data. Eventually, the selected route will converge to an optimal route.

4.3 On-the-fly density collection

As stated above, the on-the-fly density collection process is done while data packets are being forwarded. Before transmitting data packets, every forwarder adds into the packets its local density information that is obtained through received beacon messages. Then, the total density of a road segment can be obtained at the end of the road segment. When packets reach the destination, density data for all road segments along the path are collected.

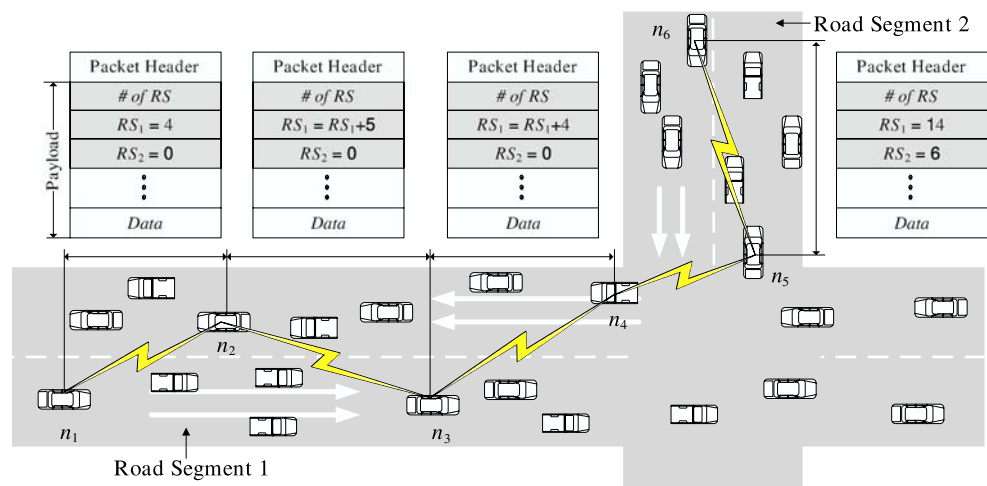
As shown in Fig. 4, the data packet of ACAR protocol is composed of two parts: packet header and

data payload. At the beginning of data payload, there are some reserved fields (bytes) for one-the-fly density collection. The first byte, denoted by N_r , records how many road segments are on the selected route. The subsequent N_r bytes record density data of all road segments on the route. The initial value of these fields is 0. Since the source node is able to compute the entire route based on historical density data from digital maps, it is easy to get the number N_r .

We now describe how a node collects the local density information and updates the corresponding byte in data packets. Since every node periodically beacons its location, velocity and id to neighbors, a node can obtain the number and positions of its one-hop neighbors. Therefore, it is easy for a node to determine whether a neighbor is in front of or behind it. For example, node n_2 in Fig. 4 infers that four nodes (including n_1) are in front of it and five nodes in the rear. Suppose node n_1 is the current packet forwarder which is at the beginning of road segment 1, and its next hop is n_2 . Before n_1 sends data packets, it adds the number of nodes between itself and n_2 (including itself) to the field RS_1 and forwards the packets to n_2 . Then, n_2 follows the same strategy and sends the packets to n_3 . Node n_3 modifies RS_1 again by adding its collected local density information, and sends out packets. Finally, packets reach the end of road segment 1 at node n_4 .

Node n_4 will decide if its next hop is still on the same road segment. If so, it continues the same procedure as node n_3 did. Otherwise, it adds 1 to the field RS_1 as it is also on road segment 1, and forward packets to its next hop (e.g. n_5 in Fig. 4). Consequently, node n_5 adds 6 to field RS_2 and forwards packets to n_6 . In the same way, when the packets reach the destination, density of every road segment on the route is collected.

Fig. 4 On-the-fly density collection mechanism



After on-the-fly density collections, the destination node needs to notify the source if there are significant discrepancies between statistical and real-time density data. If so, the source node recalculates the routes with newly collected density information; otherwise, the same route will be used for future packets.

4.4 Next hop selection

On each road segment in the selected route, packets may be forwarded through multiple hops from the beginning to the end of the road segment. The next hop will be selected using a metric that minimizes the PER of route on each road segment. The PER of a link between two nodes can be calculated by counting the number of successfully delivered packets and dropped ones. This is calculated during the beacon period and thus does not incur additional network overhead.

The original geographic routing protocols [18, 24] choose the farthest node as the next hop, since this selection can minimize the total number of hops to the destination. However, the link quality to the farthest node is usually weak because PER increases as the transmission distance increases. However, selecting next hop with a shorter distance will increase the number of hops. As proven in [13], the data delivery ratio will decrease as the hop number increases. So there is a trade-off between shorter transmission distance and smaller number of hops.

To address this issue, every node needs to measure the packet error rate of all neighbors. Suppose on a road segment there are two neighboring node n_a and n_b , and they periodically send their locations to each other. By counting the number of packets successfully delivered and dropped, the expected transmission count (ETX) can be calculated using the approach in [6]. Then the PER from n_a to n_b is obtained as:

$$PER_{ab} = 1 - \frac{1}{ETX_{ab}} \tag{40}$$

where ETX_{ab} is the expected transmission count from node n_a to n_b . In the same way, PER_{ba} can be computed. Since the route is already known (stored in the packet header), node n_a then computes the remaining distance (denoted as Dis) from itself to the next intersection. Suppose the distance between node n_a and n_b is d , then the PER of the remaining route on this road segment can be estimated by:

$$PER = 1 - (1 - PER_{ab})^{\lceil \frac{Dis}{d} \rceil} \tag{41}$$

We assume different parts of the same road segment have the similar communication environment, thus the

distance between nodes will be the dominant factor that affects the data delivery ratio. So among its neighbors, node n_a selects the one that minimizes the PER of the remaining path as the next hop. The same next hop selection will be done on all following road segments aiming to achieve the highest data delivery ratio along the whole route. However, due to frequent network partitions in VANETs, a data forwarder may have no neighbors in the forwarding direction. In these cases, we adopt the carry and forward scheme [4] that buffers packets and waits until there exists an available next hop. Then the packet will be fetched from the buffer and forwarded again.

5 Simulations and results

5.1 Mobility of nodes

Since modeling of complex vehicle movement is important for accurately evaluating protocols, we generated the movement of nodes using VanetMobiSim [14] whose mobility patterns have been validated against TSIS-CORSIM, a well-known and validated traffic generator. The VanetMobiSim features new realistic automotive motion models at both macroscopic and microscopic levels, and also supports traffic lights, lane changes and speed regulations.

We compared the network connectivity model with data collected through VanetMobiSim simulations for a set of parameters: length of road segment, average vehicle velocity and traffic light period. Specifically, as shown in Fig. 5, there are 7 road segments (each is 1000 m) in the map, the average velocity of vehicles is 10 m/s and the traffic light period is 120 s. Those small squares denote vehicles moving on the road, the number besides them are the node IDs. Our goal is to collect the network connectivity and density information on

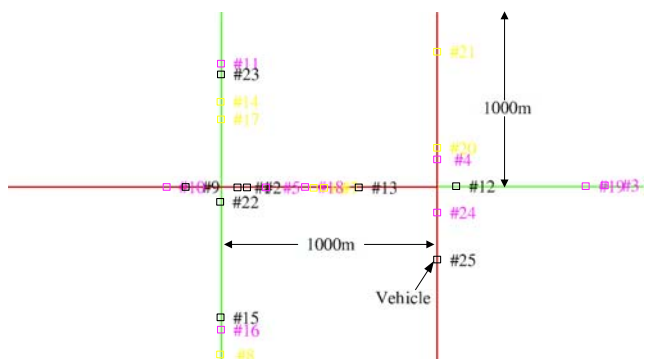


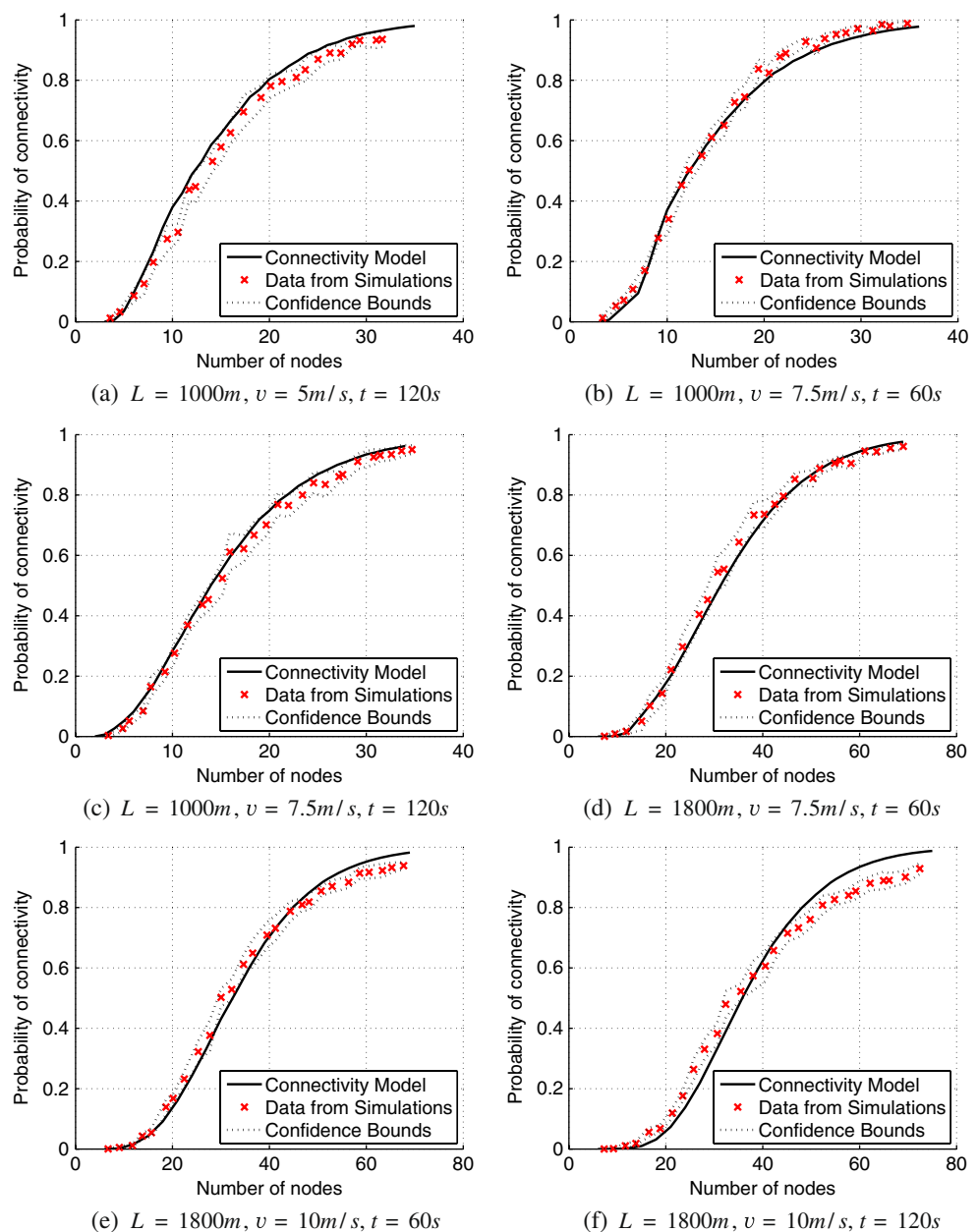
Fig. 5 A VanetMobiSim snapshot of nodes movements

the middle road segment ending with two traffic lights. The simulation time is 2000 s and we check every second if the network is connected. The number of times that networks are connected is denoted as t_c and the probability of network connectivity can be calculated as $t_c/2000$. Similarly, the average network density can be collected, though it may not be an integer. We repeated the same scenario 10 times with 10 different random seeds to achieve a high confidence level. As shown in Fig. 6a–f, with different road lengths, velocities and traffic light periods, the connectivity model matches

the value obtained from VanetMobiSim very well (confidence level is 95%).

In the above simulations, there is only one road segment containing two lanes in each driving direction. We also verified the connectivity model in the cases of more lanes (e.g. 3–5 lanes), one traffic light at the end of a road segment and routes consisting of multiple road segments. The results showed our connectivity model matched the simulation results very well. However, due to space limitation those results are omitted in this paper.

Fig. 6 Validation of the connectivity model with data generated by VanetMobiSim (a–f)



5.2 Digital map

We used two maps in simulations to show the high performance of ACAR, and how different network densities and vehicles velocities affect this protocol.

One map is illustrated in Fig. 1, which contains 5 major road segments: $I_A I_B$, $I_A I_C$, $I_A I_D$, $I_B I_C$ and $I_C I_D$. The length of each road segment and number of nodes deployed on them are the same as we described in Section 3.1. With this scenario, we evaluate the basic network performance of ACAR such as: data delivery ratio, end-to-end delay and throughput.

Within a $1000m \times 1000m$ area, street layout of the second map is loaded from the topologically integrated geographic encoding and referencing (TIGER) database, which is used by the United States census bureau to describe land attributes of U.S. The map from a city in Tennessee, centered at latitude 35162102 and longitude -84877562 , has 15 intersections and 38 road segments as shown in Fig. 7. We use this map to evaluate how network density and vehicle velocity impact the performance of ACAR.

5.3 Network simulation

We simulated the ACAR protocol in NS2 (ns-2.29) and compared it with VADD [41], MOVE [19], GPSR* and GSR*. The original GPSR [18] and GSR [24] simply drop packets when network disconnections occur, so we add carry-and-forward schemes in them and named them as GPSR* and GSR*, respectively. To make fair comparisons between ACAR and other trajectory based routing protocols, we also implemented the neighbor location predication scheme on VADD and GSR*.

Because the proper PHY/MAC modules for vehicular communications are still under development and not available for NS2, we adopt the channel fading

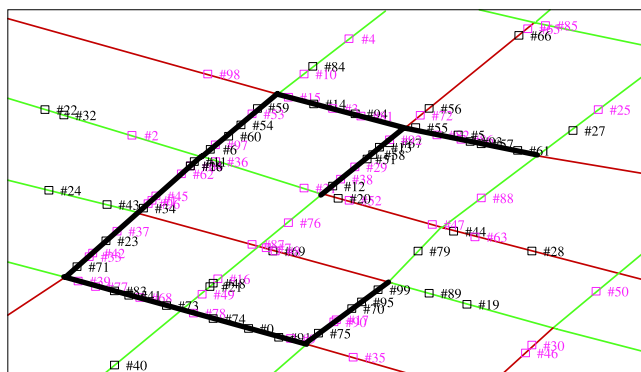


Fig. 7 Street layout in the simulation area

Table 1 Simulation parameters

| Parameter | Value |
|--------------------------|-----------------------|
| Number of lanes | 2 lanes per direction |
| Number of nodes | 40–200 |
| Velocity | 10–90 miles/hour |
| Period of traffic lights | 60 s |
| Communication range | 250 m |
| Beacon interval | 1.0 s |
| Buffer size | 64 KB |
| Packet size | 512 Bytes |

model proposed in [39] and IEEE 802.11a as the MAC/PHY protocol. Since this paper focuses on evaluating the network performance of all protocols, we omit the exact simulation of lower layers but consider it in our future work when IEEE standards for vehicular communication are finalized. Details of simulation parameters are listed in Table 1.

We first simulated the scenario shown in Fig. 1. In the simulations, different data sending rates (1 to 10 pkts/s) were used to evaluate the data delivery ratio, end-to-end delay and network throughput. A source node is randomly selected to communicate with a fixed destination. Given a real-time location service, ACAR works well if the destination is mobile. However, we considered a fixed destination to model applications described in Section 1. The simulation time is 2000 s and each scenario is repeated 20 times to achieve results with a high level of confidence.

5.4 Data delivery ratio

Data delivery ratio is the number of received packets at the destination divided by the total number of packets sent into networks. As shown in Fig. 8, ACAR achieves

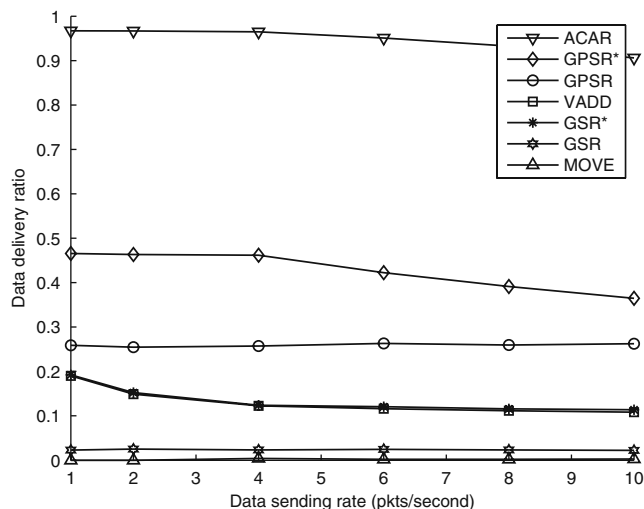


Fig. 8 Data delivery ratio of the scenario shown in Fig. 1

the highest data delivery ratio (above 90%). This is because ACAR forwards packets along route on road $I_A I_B I_C$ with the highest transmission quality.

As shown in Fig. 8, GPSR* and GPSR give the second and third highest data delivery ratios, respectively. When network partitions happen, GPSR and GPSR* utilize perimeter mode searches to find routes, so packets may finally delivered on road $I_A I_B I_C$ which has the highest transmission quality. However, GPSR* only successfully delivered about half of packets compared to the performance of ACAR. This is because, after packets are forwarded on road $I_A I_C$ or $I_A I_D$, it is possible that there are no connected links back to road $I_A I_B$. So these packets are buffered and carried by nodes moving on road $I_A I_C$ or $I_A I_D$. On the other hand, wireless transmission qualities of these two roads are very bad, so the data delivery ratio of GPSR* forwarding packets along them is very low. Since we implemented the carry-and-forward scheme on GSPR*, it delivered 10–20% more packets than GPSR. So we conclude that the carry-and-forward scheme is very helpful for routing protocols in VANET to achieve high data delivery ratios.

GSR* selects road $I_A I_C$ to forward packets, as it is the geographic shortest path to the destination. According to the connectivity model in VADD, path $I_A I_C$ provides the shortest delivery delay, so it is chosen to route packets. However, the connectivity probability of this road is just .29, and the wireless transmission quality is even lower. Therefore, the overall data delivery ratio of packets being routed on this road is very low. Since GSR* and VADD choose the same path for routing, they generate very similar data delivery ratio results.

The original GSR protocol gives a lower data delivery ratio (only .02), compared to the extended version GSR*. This is because on GSR*, we implemented NLP and carry-and-forward mechanisms. The NLP scheme can help nodes to correctly select the next hop and the carry-and-forward scheme can avoid packet loss due to network partitions. The data delivery ratio of GSR* is about 5–10 times that of GSR. Therefore, we conclude the NLP mechanism is also necessary for VANET routing protocols to achieve high data delivery ratios.

MOVE protocol delivered the least number of packets in our simulations. In MOVE, there are 7 forwarding rules being used to select the next hop. If none of neighbors satisfies these forwarding rules, packets will be carried by current node. So packets are more likely to be buffered and carried by vehicles instead of being greedily sent out. As we will describe later, these packets may be dropped due to packets expiration, weak wireless links to next hops and buffer overflows. The number of packet loss due to these reasons is very

high for MOVE, so it gives the lowest data delivery ratio compared to others.

5.5 Reasons of packet loss

There are mainly three reasons of packet loss for all VANET protocols: packets expired, weak wireless links and buffer overflow. We measured the number of lost packets due to each reason, and then find the major cause of packet loss for each protocol.

5.5.1 Expired packets

Since we cannot run simulations an infinite number of times, when simulations are terminated, there might be some packets, called expired packets, still in buffers and these packets will be dropped due to their huge delays. As shown in Fig. 9, the fraction of expired packets of MOVE is almost 5–6 times that of the others. However, ACAR, VADD, GSR* and GPSR* have the similar number of expired packets. The reason is that, in ACAR, VADD, GSR* and GPSR* protocols, packets are greedily forwarded to the next hop; but in MOVE, if none of the neighbors satisfies the forwarding rules (7 rules), packets will be carried by the current node. Therefore, packets will be more likely to be buffered in MOVE than the others. However, due to the small number of expired packets, we conclude that packet expiration is not the dominant reason for packet loss.

5.5.2 Wireless transmission loss

Packet loss can also be caused by weak wireless links to next-hop nodes, e.g. the next hop is too far away or even out of the communication range of current packet

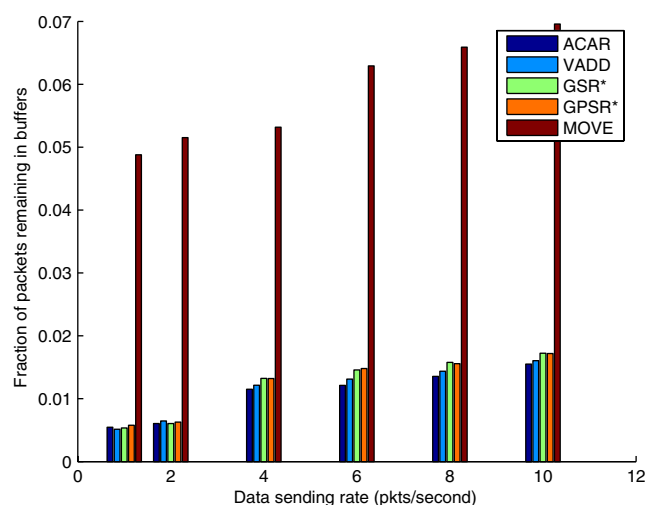


Fig. 9 Fraction of packets still in buffers

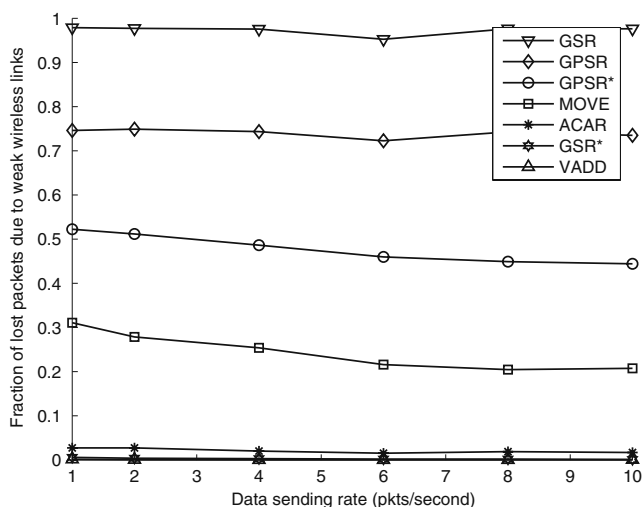


Fig. 10 Fraction of packets dropped in wireless transmissions

forwarder. As shown in Fig. 10, the number of this type of packet loss is much higher than that of expired packets. In Fig. 10, we note the original GSR has about 95% packets dropped due to this reason. GSR chooses nodes on road $I_A I_C$ to forward packets, but the probability of network connectivity on this road is so low that most packets are dropped because there is no available next hop. The original GPSR also suffers from this problem because not all packets can be routed along road $I_A I_B I_C$, i.e. some packets are dropped on road $I_A I_C$ or $I_A I_D$ before they are forwarded back to $I_A I_B I_C$ through perimeter searches. However, GPSR* can reduce this kind of packet loss. Because if there is no available next hop, packets are not simply dropped but buffered and sent when a next hop becomes available. Since GPSR* does not have the NLP mechanism, most packets dropping in GPSR* is caused by the problem of out-of-date neighbors.

MOVE gives a smaller number of packet loss in this case because packets will be most likely buffered instead of being greedily sent out. Since NLP mechanism is implemented on both VADD and GSR*, they have fewer packets dropped for this reason. In addition to NLP, ACAR carefully selects every next hop and therefore, gives the lowest packet loss due to weak wireless links.

In summary, we can conclude that weak wireless link is the major reason for packet loss in GSR, GPSR and GPSR*.

5.5.3 Buffer overflow

Another reason of packet loss in networks is buffer overflow. Figure 11 presents the percentage of lost

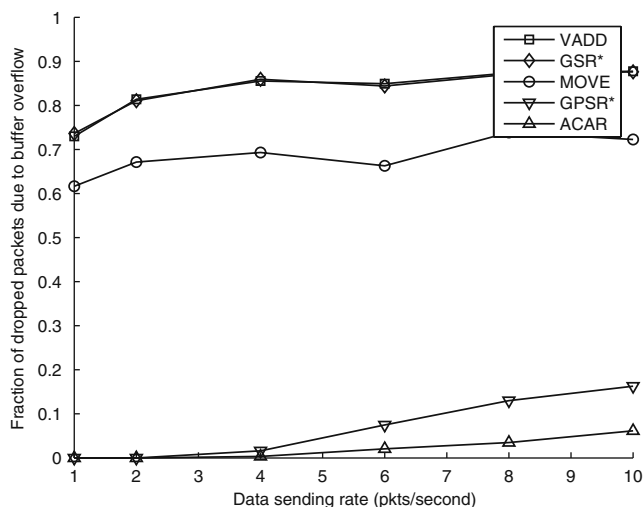


Fig. 11 Fraction of packets dropped due to buffer overflow

packets due to this problem. As shown in the figure, VADD and GSR dropped more than 70% packets due to this reason. Therefore, if the size of buffer is large enough so that all packets can be buffered and carried by vehicles, VADD and GSR can give a similar data delivery ratio as that of ACAR. In other words, ACAR has a lower requirement of the capacity of buffer on vehicles to achieve a high data delivery ratio. As mentioned before, because of the 7 rules for selecting the next hop, most packets will be buffered by MOVE. So we can see from Fig. 11, more than 60% of the packets are dropped in MOVE because it has already buffered too many packets and had no space in the buffer for more packets.

Compared to the packet loss caused by weak wireless links, buffer overflow problem is not a big issue for GPSR*, but is a significant one for ACAR. Therefore, we conclude buffer overflow is the major reason of packet loss for GSR*, VADD, MOVE and ACAR.

5.6 End-to-end delay

The end-to-end delay is defined as the average time taken for a packet to be transmitted across networks from source to destination. As shown in Fig. 12, MOVE gives the largest end-to-end delay, which is mainly because of the long time vehicles carry packets. There are 7 forwarding rules in MOVE which determine if packets are transmitted from current node to the next hop. Even though a neighbor is closer to the destination, it may not satisfy the forwarding rules and thus cannot relay packets. Therefore, more packets will be put into buffers and that results in a larger delivery delay since

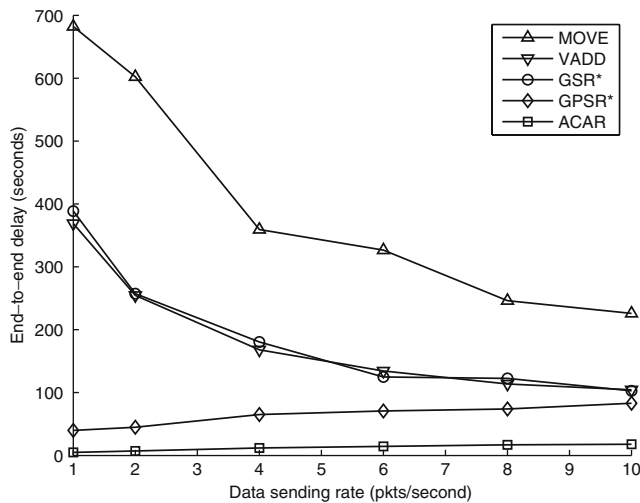


Fig. 12 End-to-end delay of the scenario shown in Fig. 1

the velocity of vehicles is much lower than the wireless transmission speed.

VADD and GSR* give a similar end-to-end delay because they select the same road segment $I_A I_C$ (in Fig. 1) to forward packets. However, the probability of network connectivity on this road is very low; thus, most packets are buffered as there is no next hop available. Since the velocity of vehicles is much slower than the speed of wireless transmission, VADD and GSR* generate a larger delay compared to ACAR and GPSR* which use connected routes on $I_A I_B I_C$ to forward packets.

An interesting observation is that when the data sending rate increases from 1 to 10 pkts/s, the end-to-end delay of MOVE decreases from about 700 to 260 s, and VADD or GSR* decreases from about 400 to 100 s. The reason of this huge reduction is: when the data sending rate is increased, more packets will be forwarded without being buffered. For example, suppose the network on $I_A I_C$ is disconnected during $[0.0, .5)$ seconds and connected within $[.5, 1.0]$ seconds. When the sending rate is 1 pkt/s, the first packet will be buffered. However, if the sending rate is 10 pkts/s, the last 5 packets are delivered without being buffered. Therefore, the average end-to-end delay of 10 pkts/s sending rate will be lower than that of 1 pkt/s case.

ACAR gives the lowest end-to-end delay, since packets are forwarded along the path $I_A I_B I_C$. There are 22 nodes on this road segment, so the probability of network connectivity of it is very high (.84). That also means most packets are delivered to the destination without being buffered, and thus ACAR saves the total time of delivering packets from the source to destination. GPSR* generates a higher end-to-end delay than

that of ACAR because some packets are forwarded to road $I_A I_C$ and $I_A I_D$, then they are detoured to road $I_A I_B I_C$ by perimeter searches. So the longer delay of GPSR* is caused by the longer route. However, it still gives a lower delay compared to VADD, GSR* and MOVE.

Unlike VADD, GSR* and MOVE, the end-to-end delay of ACAR and GPSR* increases as the data sending rate increases, due to two reasons. Firstly, when more packets are injected into the networks, the probability of packet collision is larger and thus the transmission delay increases. Secondly, higher network traffic increases the queuing time on each forwarder (vehicle), and also the end-to-end delay.

5.7 Network throughput

We compared throughputs of MOVE, GPSR*, GSR*, VADD and ACAR in the network shown in Fig. 1. Results in Fig. 13 show that ACAR outperforms all the other protocols, i.e. it achieves the highest network throughput of 84 kb/s. This value is about three times that of GPSR* which is second best protocol. Since packets are forwarded along a route with the highest transmission quality in ACAR, link quality per hop is higher than that of others. Therefore, the data delivery ratio and end-to-end delay can be improved. We also note the shapes of GPSR and GPSR* results are very similar to that of ACAR because both GPSR and GPSR* delivered most packets along the route on $I_A I_B I_C$, which is the route chosen by ACAR too.

VADD and GSR* give the similar throughput as the data sending rate increases because they all chosen

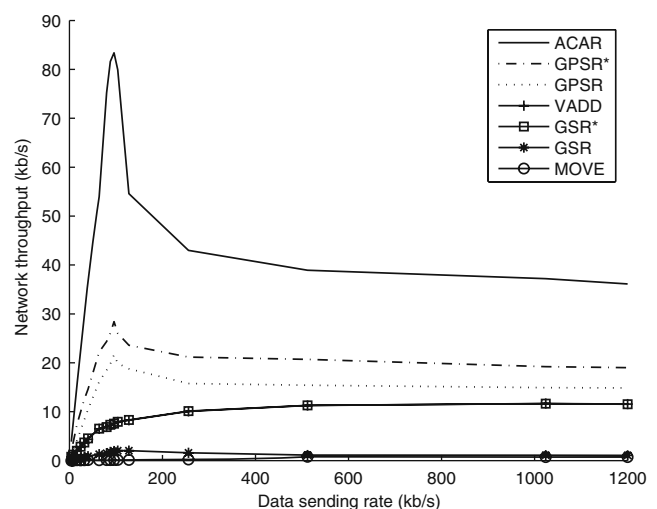


Fig. 13 Network throughput of the scenario shown in Fig. 1

routes on the same road segment ($I_A I_C$) to deliver packets. Since the probability of connectivity of road $I_A I_C$ is low, the throughput of VADD and GSR* is lower than that of ACAR, GSR* and GPSR.

An interesting observation of VADD and GSR* is that their throughputs increase when the data sending rate increases from 1 kb/s to 500 kb/s and become stable after that. This is quite different from ACAR and GPSR*, whose throughputs decrease after reaching the peak values. As mentioned previously, when the data sending rate increases, the chance of packets being delivered increases and so does the network throughput. However, if the data sending rate is so high that buffer overflows happen on nodes, the larger data sending rate is not helpful for network throughput. At this point, every node will periodically send out one packet from its buffer when a next hop is available. Since the time interval for periodic buffer checking is a fixed value, the network throughput becomes stable in this situation.

GSR gives the second lowest throughput performance because packets will be simply dropped when it faces network disconnections which is very common on road $I_A I_C$. MOVE will choose nodes on $I_A I_D$ and $I_C I_D$ to forward packets, so the overall network throughput will be very low due to low network connectivity and longer delivery path.

5.8 Impact of network density

To evaluate the network performance of ACAR in a more general case, we simulated networks in the second map with data from the U.S. TIGER database. We evaluate how different network densities affect the network performance, in terms of data delivery ratio and end-to-end delay.

In reality, vehicles are not evenly distributed on roads, so we manually deploy more vehicles (70% of the total number) on certain roads which are highlighted by bold lines in Fig. 7, and fewer nodes (30%) on the others. The total number of nodes in networks varies from 40 to 200. Since vehicles can only move on roads instead of the entire simulation area, we define *network density* as the ratio between number of nodes and the total length of all road segments. The total length of roads in the map is 7878m, so the network density varies from 1/197 to 1/40 nodes per meter.

As shown in Fig. 14, except for VADD and MOVE, all protocols deliver more packets as the network density increases. This is because when the number of nodes increases, the expected network connectivity probability increases too and so does the data delivery ratio. From Fig. 14, we note that when the network

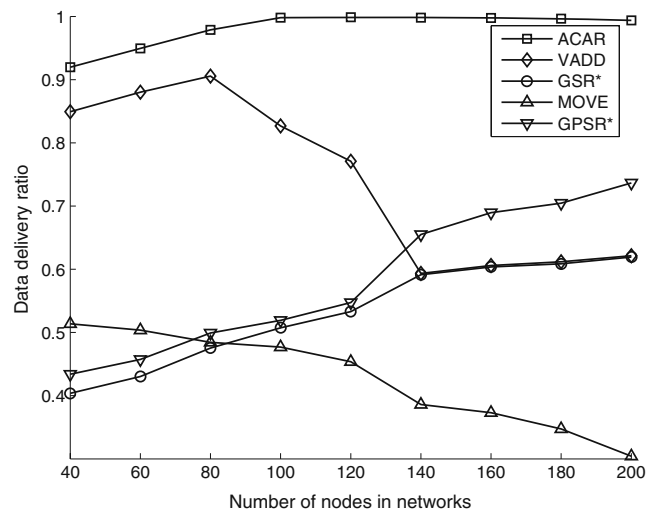


Fig. 14 Data delivery ratio vs number of nodes

density is low, 40 to 120 nodes, GPSR* and GSR* give similar data delivery ratios. That means the perimeter search in GPSR* cannot drastically improve the data delivery ratio when network density is low, but it does help to reduce the end-to-end delay as shown in Fig. 15. However, when the number of nodes is larger than 120, the network connectivity probability increases. Then, it is more likely for GPSR* to find a connected path instead of forwarding packets on the geographic shortest path. Therefore, it delivered more packets than GSR* which still forwards packets on the geographic shortest road segments.

In the MOVE protocol, the larger the network density, the higher the probability of Ping-Pong situation may occur (as described in Section 3.1). So the delivery

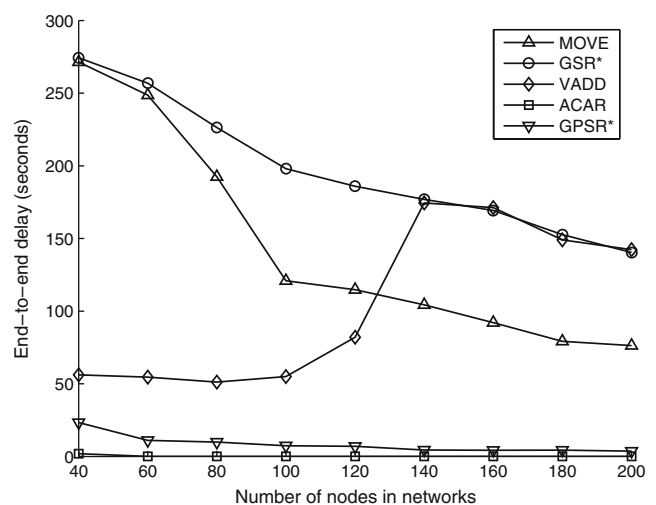


Fig. 15 End-to-end delay vs number of nodes

ratio of MOVE is reduced. For VADD, its data delivery ratio increases when network density is low (40–80 nodes), decreases when network density is medium (80–140 nodes), and slightly increases when network density is large (140–200 nodes). When the network density is low, VADD considers no connected network on most road segments, and will forward packets along the path with higher probability of connectivity, e.g. roads marked by bold lines in Fig. 7. Thus, the delivery ratio will increase when more nodes are deployed in networks. However, when the network density becomes larger, VADD may find there are some connected networks on other roads which are closer to the destination. Then, VADD will forward packets along those roads. Due to the limitation of connectivity model in VADD, probabilities of connectivity of these roads are actually very low. So the data delivery ratio decreases until these roads are really connected (140 nodes). After that, the data delivery ratio of VADD is similar to that of GSR* because both of them will choose the geographic shortest path to forward packets. The delivery ratio slightly increases when more nodes are deployed on the shortest path.

The end-to-end delay of all protocols, except for VADD, drops when network density increases. This is because network connectivity probability increases when more nodes are deployed in networks. As shown in Fig. 15, ACAR and GPSR* give the lowest and second lowest end-to-end delay, respectively. Since ACAR forwards packets along routes with the highest transmission quality, the number of buffered packets (during network disconnections) is less often than that of GPSR*, resulting in a lower delay. On the other hand, when network disconnections happen, GPSR* in the perimeter mode can search for another connected path (e.g. the path used by ACAR), so it also generates a small delay compared to others.

GSR* only forwards packets along pre-defined routes, i.e. the geographic shortest path, so it has no opportunity to find a better connected path as GPSR* does. Therefore, it gives a much higher end-to-end delay. As mentioned above, in MOVE, nodes will carry more packets in their buffers and this will reduce the data delivery rate. Thus, MOVE gives a very high end-to-end delay.

An interesting observation of VADD's end-to-end delay is: it decreases from 40 to 80 nodes, and increases from 80 to 140 nodes, and decreases again from 140 to 200 nodes. The reason is similar to that of the delivery ratio results: with the connectivity model used in VADD, some disconnected paths are considered connected and selected as routes to forward packets. Along those frequently-disconnected paths, packets are

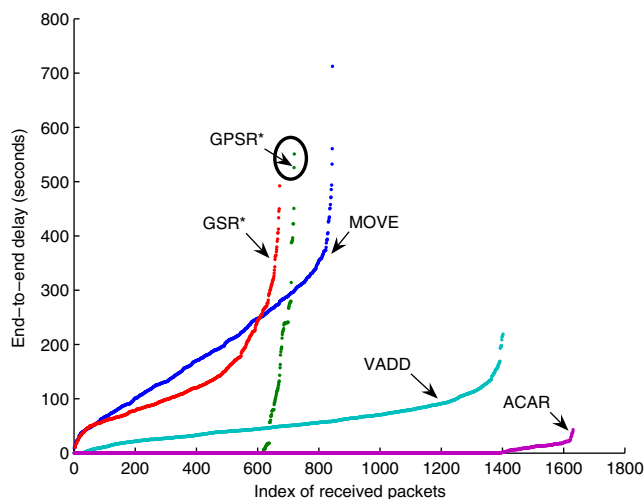


Fig. 16 Delay distribution of received packets with 40 nodes in the networks

frequently buffered so the average end-to-end delay of VADD is higher than GPSR* and ACAR.

As the end-to-end delays of ACAR and GPSR* are very similar, we further investigate the delay distribution of delivered packets. For example, when there are 40 nodes in networks, the delay distribution of received packets for all protocols is shown in Fig. 16. The x-axis denotes indices of the received packets, and y-axis for end-to-end delays which are measured in seconds. We order received packets by their end-to-end delays. Dots denote the end-to-end delays of corresponding packets. As we can see, ACAR delivers most packets with smaller delays; while in GPSR*, some delivered packets have very large delays (a few hundred seconds). In addition, although GPSR* and GSR* deliver similar numbers of packets, GPSR* definitely routes packets along faster but longer paths than those used by GSR*. Some packets (1st to 600th) in GPSR* are delivered successfully along connected paths, while others (after 600th) are buffered and carried by vehicles. Since paths selected by GPSR* are longer, delays of some packets circled in Fig. 16 are even larger than those of GSR*. In summary, we conclude that ACAR not only gives the lowest average delay but also delivers more packets with smaller delays compared to other protocols.

5.9 Impact of velocity

Since mobility of nodes may affect the performance of protocols, we simulated networks with 100 nodes moving with different velocities. As shown in Fig. 17, when networks become more dynamic, the data delivery ratio decreases for all protocols. However, ACAR

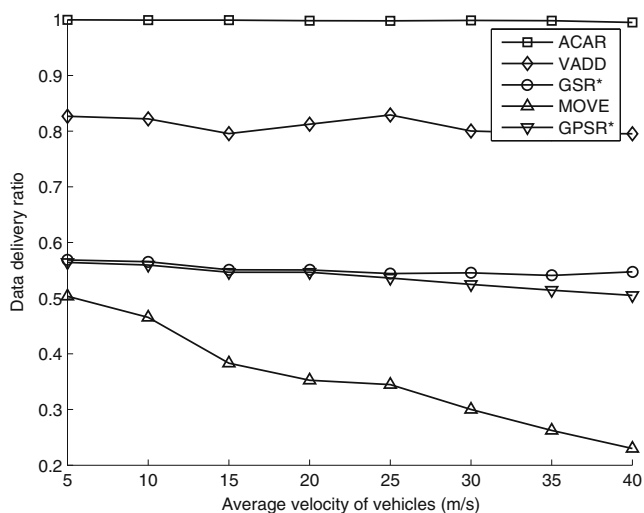


Fig. 17 Data delivery ratio vs different velocities (100 nodes)

is only slightly affected (reduced by 1%) by the change of node mobilities. This is because higher velocity does not affect our connectivity model but only the choice of each next hop. In fact, the larger the velocity, the lower the accuracy of predicting neighbors positions.

Since we implemented NLP on VADD and GSR*, their data delivery ratios drop more slowly than GPSR* and MOVE. For GPSR*, as no NLP algorithm is available, it may select next hops which are already out of the communication range due to the high speed of its neighbors. The situation is even worse for MOVE protocol. Unlike other protocols in which packets are routed on either geographic shortest paths or high connectivity paths, MOVE forwards packets to nodes moving towards the destination. However, this node

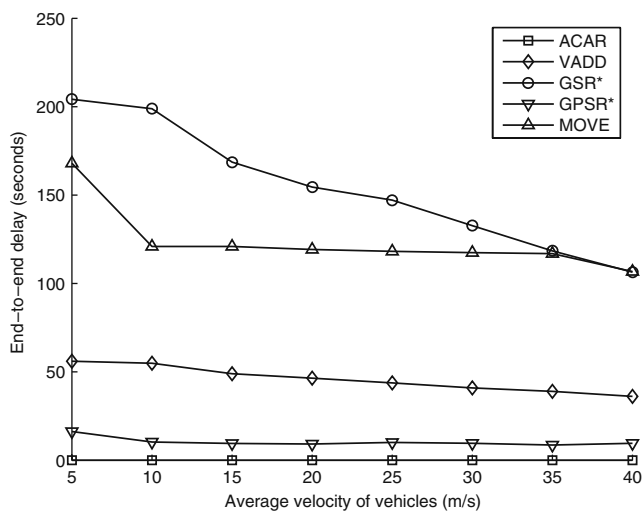


Fig. 18 End-to-end delay vs different velocities (100 nodes)

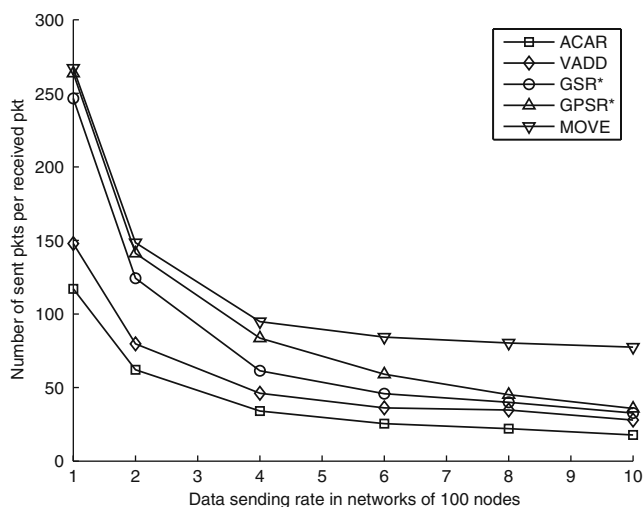


Fig. 19 Number of packets sent in networks per delivered packet (100 nodes)

may move away from the destination a few seconds later. If no next hop is available, which is very common for MOVE, current forwarder (carrying packets) will move away from the destination very fast and so extends the total routing path. The longer the route, the higher is the chance of wrongly selecting next hops. So the delivery ratio of MOVE drops very fast when the velocity increases.

As shown in Fig. 18, the end-to-end delay of ACAR is very low because ACAR forwards packets on routes with the highest transmission qualities. So the delay of ACAR is mainly composed of wireless transmission and protocol queuing delays, which are very small.

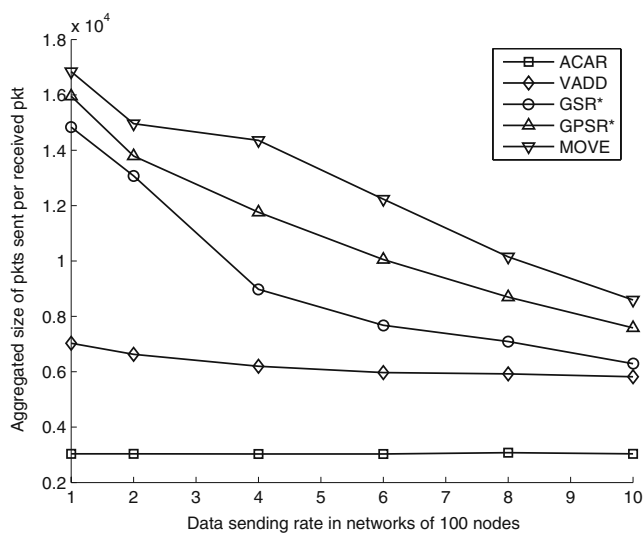


Fig. 20 Size of packets sent in networks per delivered packet (100 nodes)

Since GPSR* utilizes the perimeter mode to find connected paths, its delay is also very low. However, end-to-end delays of VADD, GSR* and MOVE are much higher and drop when the velocity increases. Because VADD, GSR* and MOVE have no mechanisms for selecting connected paths, their delays are higher due to more packets being buffered. In addition, when the velocity increases, packet carriers can move faster to the destination and thus decrease the average end-to-end delay.

5.10 Networking overhead

Networking overhead is defined as the number of packets sent into networks for every delivered packet. In other words, it is the ratio between the number of sent packets (beacon and data messages) and the number of received packets. As every node sends beacon messages periodically, e.g. every 1 s, this kind of packets make up the majority of networking overhead. When the data sending rate increases, more packets will be delivered to the destination, so the overall networking overhead decreases. The total number of sent packets for all protocols are similar, so the networking overhead of ACAR is the lowest as it delivers more packets than others (Fig. 19).

In ACAR, there is an on-the-fly density collection scheme which will increase the size of every forwarded packet. So we further evaluate the networking overhead by investigating the total size of packets sent into networks per delivered packet. As the periodic beacon scheme is the same for all protocols, we only consider the size of data packets here. As shown in Fig. 20, ACAR gives the lowest networking overhead in terms of the average size of data messages per delivered packet. The major reason is that ACAR delivers more packets than others, so reduces the overall networking overhead.

6 Conclusion

We have presented an algorithm for adaptively selecting routes based on statistical and real-time data to avoid the influence of inaccurate statistical density data. Because the selected path has the best transmission quality, ACAR achieves a higher successful data delivery ratio and lower end-to-end delay compared to others. Moreover, since the route length can be calculated before forwarding packets, every next hop is selected by minimizing the packet error rate of the entire path. Our simulation results show that ACAR is much more suitable for VANET than other protocols

because of its higher data delivery ratio, throughput and lower networking delay. In addition, it works very well even when the statistical data of road density is not accurate. We will investigate in our future work how we can further improve the protocol's performance by integrating traffic load with the quality model.

References

1. Ahmed S, Kanere SS (2006) SKVR: scalable knowledge-based routing architecture for public transport networks. In: VANET '06: proceedings of the 3rd international workshop on vehicular ad hoc networks. ACM, New York, pp 92–93
2. Balasubramanian A, Mahajan R, Venkataramani A, Levine BN, Zahorjan J (2008) Interactive wifi connectivity for moving vehicles. *SIGCOMM Comput Commun Rev* 38(4): 427–438
3. Bernsen J, Manivannan D (2009) Unicast routing protocols for vehicular ad hoc networks: a critical comparison and classification. *Perv Mob Comput* 5(1):1–18
4. Chen ZD, Kung H, Vlah D (2001) Ad hoc relay wireless networks over moving vehicles on highways. In: *MobiHoc '01: proceedings of the 2nd ACM international symposium on mobile ad hoc networking & computing*. ACM, New York, pp 247–250
5. Costa P, Frey D, Migliavacca M, Mottola L (2006) Towards lightweight information dissemination in inter-vehicular networks. In: *VANET '06: proceedings of the 3rd international workshop on Vehicular ad hoc networks*. ACM, New York, pp 20–29
6. Couto DSJD, Aguayo D, Bicket J, Morris R (2005) A high-throughput path metric for multi-hop wireless routing. *Wirel Netw* 11(4):419–434
7. Ding Y, Wang C, Xiao L (2007) A static-node assisted adaptive routing protocol in vehicular networks. In: *VANET '07: proceedings of the fourth ACM international workshop on vehicular ad hoc networks*. ACM, New York, pp 59–68
8. Dousse O, Thiran P, Hasler M (2002) Connectivity in ad-hoc and hybrid networks. In: *INFOCOM 2002. Proceedings IEEE 21st annual joint conference of the IEEE computer and communications societies, vol 2*. IEEE, Piscataway, pp 1079–1088
9. Eriksson J, Balakrishnan H, Madden S (2008) Cabernet: vehicular content delivery using WiFi. In: *MobiCom '08: proceedings of the 14th ACM international conference on Mobile computing and networking*. ACM, New York, pp 199–210
10. Fiore M, Härri J (2008) The networking shape of vehicular mobility. In: *MobiHoc '08: proceedings of the 9th ACM international symposium on mobile ad hoc networking and computing*. ACM, New York, pp 261–272
11. Fiore M, Harri J, Filali F, Bonnet C (2007) Vehicular mobility simulation for VANETs. In: *ANSS '07. 40th Annual, Simulation symposium, 2007*, pp 301–309
12. Grossglauser M, Tse D (2002) Mobility increases the capacity of ad hoc wireless networks. *IEEE/ACM Trans Netw* 10(4):477–486
13. Gupta P, Kumar P (2000) The capacity of wireless networks. *IEEE Trans Inf Theory* 46(2):388–404
14. Härri J, Filali F, Bonnet C, Fiore M (2006) VanetMobiSim: generating realistic mobility patterns for vanets. In: *VANET*

- '06: proceedings of the 3rd international workshop on vehicular ad hoc networks. ACM, New York, pp 96–97
15. Helal S, Mann W, El-Zabadani H, King J, Kaddoura Y, Jansen E (2005) The gator tech smart house: a programmable pervasive space. *Comput* 38(3):50–60
 16. Ho YH, Ho AH, Hua KA (2007) Routing protocols for inter-vehicular networks: a comparative study in high-mobility and large obstacles environments. *Comput Commun* 31(12):2767–2780
 17. Johnson DB, Maltz DA (1996) Dynamic source routing in ad hoc wireless networks. In: Imielinski T, Korth H (eds) *Mobile computing*, vol 353. Kluwer, Dordrecht
 18. Karp B, Kung HT (2000) GPSR: greedy perimeter stateless routing for wireless networks. In: *MobiCom '00: proceedings of the 6th annual international conference on mobile computing and networking*. ACM, New York, pp 243–254
 19. LeBrun J, Chuah CN, Ghosal D, Zhang M (2005) Knowledge-based opportunistic forwarding in vehicular wireless ad hoc networks. In: *Vehicular technology conference. VTC 2005-Spring, 2005 IEEE 61st, vol 4*. IEEE, Piscataway, pp 2289–2293
 20. Lee SH, Lee U, Lee KW, Gerla M (2008) Content distribution in VANETs using network coding: the effect of disk I/O and processing O/H. In: *SECON '08. 5th Annual IEEE communications society conference on sensor, mesh and ad hoc communications and networks*, pp 117–125
 21. Leontiadis I, Mascolo C (2007) GeOpps: geographical opportunistic routing for vehicular networks. In: *IEEE international symposium on a world of wireless, mobile and multimedia networks, (WoWMoM 2007)*, pp 1–6
 22. Li F, Wang Y (2007) Routing in vehicular ad hoc networks: a survey. *IEEE Trans Veh Technol Mag* 2(2):12–22
 23. Liu G, Lee B, Seet B, Foh C, Lee K (2004) A routing strategy for metropolis vehicular communications. In: *Proc. international conference on information networking*, pp 533–542
 24. Lochert C, Mauve M, Füssler H, Hartenstein H (2005) Geographic routing in city scenarios. *SIGMOBILE Mob Comput Commun Rev* 9(1):69–72
 25. Mo Z, Zhu H, Makki K, Pissinou N (2006) MURU: a multi-hop routing protocol for urban vehicular ad hoc networks. In: *3rd annual international conference on Mobile and ubiquitous systems—workshops*, pp 1–8
 26. Naumov V, Gross T (2007) Connectivity-aware routing (CAR) in vehicular ad-hoc networks. In: *INFOCOM 2007. 26th IEEE international conference on computer communications*. IEEE, Piscataway, pp 1919–1927
 27. Naumov V, Baumann R, Gross T (2006) An evaluation of inter-vehicle ad hoc networks based on realistic vehicular traces. In: *MobiHoc '06: proceedings of the 7th ACM international symposium on mobile ad hoc networking and computing*. ACM, New York, pp 108–119
 28. Perkins C, Royer E (1999) Ad-hoc on-demand distance vector routing. In: *Proceedings, WMCSA '99. 2nd IEEE workshop on mobile computing systems and applications*, pp 90–100
 29. Rybicki J, Scheuermann B, Kiess W, Lochert C, Fallahi P, Mauve M (2007) Challenge: peers on wheels—a road to new traffic information systems. In: *MobiCom '07: proceedings of the 13th annual ACM international conference on mobile computing and networking*. ACM, New York, pp 215–221
 30. Santi P, Blough D (2003) The critical transmitting range for connectivity in sparse wireless ad hoc networks. *IEEE Trans Mob Comput* 2(1):25–39
 31. Seskar I, Maric S, Holtzman J, Wasserman J (1992) Rate of location area updates in cellular systems. In: *IEEE 42nd vehicular technology conference vol 2*. IEEE, Piscataway, pp 694–697
 32. Skordylis A, Trigoni N (2008) Delay-bounded routing in vehicular ad-hoc networks. In: *MobiHoc '08: proceedings of the 9th ACM international symposium on mobile ad hoc networking and computing*. ACM, New York, pp 341–350
 33. Son D, Helmy A, Krishnamachari B (2004) The effect of mobility-induced location errors on geographic routing in ad hoc networks: analysis and improvement using mobility prediction. In: *IEEE wireless communications and networking conference, WCNC*, pp 189–194
 34. Stepanov I, Hahner J, Becker C, Tian J, Rothermel K (2003) A meta-model and framework for user mobility in mobile networks. In: *ICON2003. The 11th IEEE international conference on networks*, pp 231–238
 35. Treiber M, Hennecke A, Helbing D (2000) Congested traffic states in empirical observations and microscopic simulations. *Phys Rev E* 62(2):1805–1824
 36. Wisitpongphan N, Bai F, Mudalige P, Tonguz O (2007) On the routing problem in disconnected vehicular ad-hoc networks. In: *INFOCOM 2007. 26th IEEE international conference on computer communications*. IEEE, Piscataway, pp 2291–2295
 37. Wu H, Fujimoto R, Guensler R, Hunter M (2004) MDDV: a mobility-centric data dissemination algorithm for vehicular networks. In: *VANET '04: proceedings of the 1st ACM international workshop on vehicular ad hoc networks*. ACM, New York, pp 47–56
 38. Yang Q, Lim A, Agrawal P (2008) Connectivity aware routing in vehicular networks. In: *Wireless communications and networking conference, WCNC*. IEEE, Piscataway, pp 2218–2223
 39. Zang Y, Stibor L, Orfanos G, Guo S, Reuerman HJ (2005) An error model for inter-vehicle communications in highway scenarios at 5.9GHz. In: *PE-WASUN '05: proceedings of the 2nd ACM international workshop on performance evaluation of wireless ad hoc, sensor, and ubiquitous networks*, pp 49–56
 40. Zhang Y, Zhao J, Cao G (2007) On scheduling vehicle-roadside data access. In: *VANET '07: proceedings of the 4th ACM international workshop on vehicular ad hoc networks*. ACM, New York, pp 9–18
 41. Zhao J, Cao G (2006) VADD: vehicle-assisted data delivery in vehicular ad hoc networks. In: *INFOCOM 2006. Proceedings, 25th IEEE international conference on computer communications*, pp 1–12



Qing Yang received his B.E. and M.E. degree in computer science and technology from Nankai University (China) and

Harbin Institute of Technology (China) in 2003 and 2005, respectively. He is currently a Ph.D candidate in the department of computer science and software engineering, Auburn University. He is the student member of IEEE and ACM, and has been the Vodafone Fellow from 2005 to 2008. His research interests include target tracking in distributed sensor networks, undersea sensor networks, routing in vehicular networks (VANET) and real-time embedding systems.



Alvin Lim is currently an associate professor of computer science and software engineering at Auburn University. He received his Ph.D. degree in computer science from University of Wisconsin at Madison in 1993. His research interests include self-organizing sensor networks, mobile and pervasive computing, network security, wireless networks, reliable and dynamically reconfigurable distributed systems, complex distributed systems, mobile and distributed databases, distributed operating systems, and performance measurement and analysis. He has published widely in journals and conferences in these networking and distributed systems areas. He is a subject area editor of the International Journal of Distributed Sensor Networks. His work had been supported by the National Science Foundation, the DARPA SensIT program, U.S. Air Force Research Lab and the U.S. Army.



Shuang Li is currently a PhD student in the department of Computer Science and Engineering at the Ohio State University. She received her master's degree from Auburn University and bachelor's degree from Wuhan University of Technology, China, both in Computer Science. Her research interests are wireless sensor networks, ad hoc networks.



Jian Fang is currently a doctoral student of the Department of Computer Science and Software Engineering at Auburn University. His research interests include mobile and distributed systems, target tracking, and video processing and transmission over sensor networks. He received his bachelor degree from the University of Shanghai for Science and Technology and master degree from the University of Alabama in Huntsville. He is a student member of IEEE.



Prathima Agrawal received her B.E. and M.E. degrees in electrical communication engineering from the Indian Institute of Science, Bangalore. She received her Ph.D. degree in electrical engineering from the University of Southern California in 1977. She is the Samuel Ginn Distinguished Professor of Electrical and Computer Engineering at Auburn University, Alabama. She is also the director of the Wireless Engineering Research and Education Center at Auburn. Her research interests are computer networks, mobile and wireless computing, and communication systems. Earlier, she was assistant vice president of the Network Systems Research Laboratory and executive director of the Mobile Networking Research Department at Telcordia Technologies (formerly Bellcore), Morristown, New Jersey, where she worked from 1998 to 2003. Prior to this, she was head of the Networked Computing Research Department at AT&T Lucent Bell Laboratories, Murray Hill, New Jersey, where she worked from 1978 to 1998 in various capacities. She has published over 200 papers and holds 50 patents with several more pending. She is a Fellow of the Institution of Electronics and Telecommunications Engineers, India, and a member of the ACM.