

Research Article

Routing Protocol in VANETs Equipped with Directional Antennas: Topology-Based Neighbor Discovery and Routing Analysis

Haipeng Li  and Zhen Xu 

School of Electronic and Information Engineering, Beihang University, Beijing, China

Correspondence should be addressed to Zhen Xu; xuzhen@buaa.edu.cn

Received 14 December 2017; Accepted 11 March 2018; Published 10 April 2018

Academic Editor: Leyre Azpilicueta

Copyright © 2018 Haipeng Li and Zhen Xu. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In Vehicular Ad Hoc Networks (VANETs), directional antenna is a good solution if a longer transmission distance is needed. When vehicles are equipped with directional antennas, however, complete paths from the sources to the destinations there may not exist. Epidemic routing protocol is considered as one of the well-performed routing protocols when the networks are intermittently connected but it can cause a heavy load to the network and great energy consumption to the nodes. First in this paper, we propose a novel neighbor discovery algorithm which makes nodes be able to sense the topology changes around them and arrange their directional antennas accordingly. Secondly, we propose a routing protocol which is based on the conventional epidemic routing protocol, and nodes make their routing decisions according to the information collected during the neighbor discovery process. Experimental results show that the proposed neighbor discovery algorithm has better performance especially in the scenario where the node density is low. Moreover, the matching routing protocol can effectively reduce the load of the network and successfully deliver the packets to its destination in a reasonable short delay.

1. Introduction

Directional antennas have been widely used in wireless ad hoc networks recently. It can produce higher gain, provide greater transmission range, and improve the network spatial reuse and throughput as well. Furthermore, the directional selectivity also reduces the co-channel interference of neighboring nodes, and directional antennas bring potential performance improvement to the mobile ad hoc networks. Despite the improvements brought by the directional antennas, their deployments also bring severe challenges to the nodes such as the neighbor discovery, and the network suffers from frequent disruptions, making the topology unstable, that is, no consistent paths between each pair of nodes.

Considering that, in a considerable proportion of real scenarios, the mobility of vehicles is not purely random. Vehicles show strong location and group preference. For example, for the vehicles in the earthquake relief sites, they

work at the areas which are under serious destruction and these areas are not uniformly distributed. Moreover, vehicles can change their locations from one area to another. Based on this kind of scenario, we formulate the neighbor discovery algorithm and the matching routing protocol in this paper. The main contribution of this paper is about two aspects. Firstly, we design a novel neighbor discovery algorithm for intermittent networks which are equipped with directional antennas. In the proposed neighbor discovery algorithm, nodes monitor the HELLO message receiving frequency for each sector to estimate the topology variations and adjust the steering time of each sector according to the history records maintained in their neighbor node lists. Secondly, we propose a routing protocol which is derived from the proposed neighbor discovery algorithm. In the routing protocol, source node makes its routing decisions according to the status of the nodes in its two hops. The stability of neighbor relationships, the communicating probability, and the time schedule for the

directional antennas of each participating node are all taken into consideration when the source node makes its routing decisions.

The rest of the paper is organized as follows. In Section 2, we introduce the related researches about neighbor discovery algorithms and routing protocols for opportunistic networks. In Section 3, we present the system models. In Section 4, we introduce and analyze our proposed neighbor discovery algorithm. Our proposed routing protocol is demonstrated in Section 5. Section 6 provides simulation results. Finally, we conclude the paper in Section 7.

2. Related Work

2.1. Neighbor Discovery. As we discussed in Section 1, neighbor discovery is the key initial step for establishing the connections among nodes in intermittent networks. In recent research papers about neighbor discovery protocols with pure directional antennas, we categorize them into two classes, *deterministic* and *random* protocols. In deterministic protocols [1–3], nodes steer their directional antennas in preset sequences. Deterministic protocols can guarantee the bounded neighbor discovery delay. However, most of them need time synchronized which is not practical in some applications. In random protocols [4–10], nodes randomly select a direction to transmit or receive HELLO messages. Compared to deterministic protocols, the most obvious advantage of probabilistic protocols is that they do not need to be synchronized, in which they are more robust and adaptive to complex scenarios. However, in most probabilistic or random protocols, the topology change of the network is not concerned. Reference [1] focuses on the oblivious neighbor discovery problems and has proposed an oblivious discovery protocol which achieves guaranteed oblivious discovery with order-minimal worst-case discovery delay. However, the topology change is not its concern. References [2–10] focus on the MAC protocols to achieve a better neighbor discovery performance. References [2–8] all need time synchronized to ensure their performance and their directional antennas just simply choose a direction when they run the neighbor discovery process. References [11, 12] take the topology into consideration. Reference [11] works on topology control problems. In this work, nodes track only a subset of their discovered neighbor nodes and it proposes a scheme called Di-ATC which tries to minimize the variance of the angular separation between the tracked neighbors of a node. In [12], nodes discover their neighbor nodes in one sector with increasing power so as to identify the closest nodes in the sector. However, [11] assumes that the network is connected at all times and the path stretch after topology control is low and [12] is only suitable for a static ad hoc network which is not realistic in VANETs with directional antennas equipped.

2.2. Routing Protocols. Since there are no complete paths in intermittent vehicle ad hoc networks with directional antennas, traditional end-to-end routing protocols for mobile ad hoc network (e.g., DSR, AODV) are not suitable anymore. The existing routing protocols that are concerned with intermittent networks mostly focus on the load easement while

the successful transmission is guaranteed. In [13, 14], nodes only generate one copy of the data packets to reduce the load of network. In methods [15–18], nodes selectively send the copies of packets to their neighbor nodes. In these *replication-based* methods, nodes carefully make the routing decisions according to several rules such as *utility-based* routing [15] or *probability-based* routing [17]. They assume that movement patterns of node are not purely random and that future contacts depend on the past information. Reference [16] proposes a method called *Spray and Focus* which refers to the fact that nodes firstly spray a fixed number of their data packets to the selected relay nodes and then nodes with only one copy of the packet can only forward this message further using a *single-copy utility-based scheme*. In [18], they investigate the TTL on the copy of packets in order to reduce the load of the networks. Due to the dynamic topology of VANETs, the size of HELLO packets and the mobility should be taken into consideration for the routing decisions. Reference [19] investigates the MAC layer contention issue that it proposes BBNC, a backbone based routing protocol with interflow Network Coding. The position information of each backbone node is critical when nodes in [19] make their routing decisions. Reference [20] focuses on the effect of packet size on the performance of routing protocols; it uses a fuzzy logic-based algorithm to select relay nodes and uses a Q-learning based approach to tune the fuzzy membership functions. For the VANETs where nodes are equipped with directional antennas, the dynamic topology is mostly caused by the scan pattern of the directional antennas, and this is the emphasis of this paper. For the routing protocols which are focusing on the directional antenna, such as in [21], nodes make their routing decisions based on the information about the geographical position of the participating nodes, which needs the use of GPS. Reference [22] focuses on analyzing the relationship between the throughput and the beam width of the directional antenna. In general, they rarely consider the switch pattern of the directional antenna.

3. System Model and Problem Formulation

3.1. System Model

3.1.1. Directional Antenna Model. We assume that the set of nodes in the network is denoted by \tilde{N} . Each node $i \in \tilde{N}$ is equipped with only one set of directional antennas. For node i , we approximate the directional antenna as a circular sector of angle θ_i ($0 < \theta_i \leq 2\pi$), and there are $N_i \triangleq 2\pi/\theta_i$ sectors around it, indexed clockwise from 0 to $N_i - 1$. When $\theta_i = 2\pi$, the directional antenna degenerates to an omnidirectional antenna. The set of sectors of node i is denoted by \hat{S}_i ; $S_i \in \hat{S}_i$ maintains a timer C_{S_i} . When C_{S_i} expires, the rotation mode of the directional antenna is decided by the status of the node:

- (i) Pure neighbor discovery mode (PND-M): if there is no transmission task for the node, then it switches its directional antenna randomly to the next sector.
- (ii) Data transmission mode (DT-M): if there are data packets conserved in the node and waiting to be transmitted, then the node begins to switch its directional

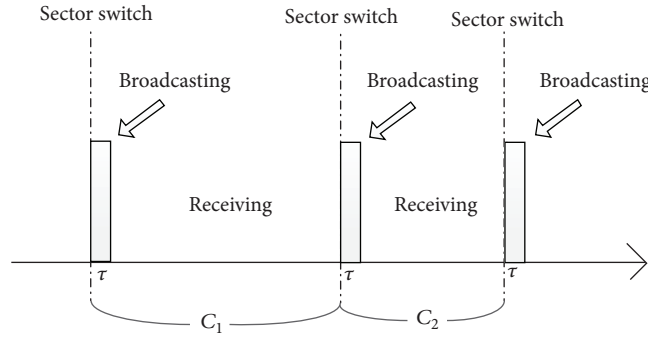


FIGURE 1: Communication model.

antenna clockwise. In this mode, the scan pattern of each node is ordered and predictable so that the routing decisions are more precise.

The initial C_{S_i} is equal for every sector and the rotation period is a constant number, T , where $T = \sum_{S_i \in \hat{S}_i} C_{S_i}$. The cover range of each node is R .

3.1.2. Communication Model. We assume that the time unit is τ , which refers to the maximal time that a HELLO message needs to be broadcasted to its neighbor successfully. As shown in Figure 1, when node i points its directional antenna to sector S_i , it broadcasts a HELLO message and then it enters the receiving state until it switches its direction antenna to the next sector. When node i receives a HELLO message from node j , then node j is a discovered neighbor node for node i . Only when two nodes point their directional antennas to the opposite direction and they are located in the transmission range of each other, the HELLO message can be received successfully. Moreover, when a node receives a HELLO message from another node, it will send back a response message conditionally. This mechanism is detailed in Section 4.

3.1.3. Mobility Model. In this paper, groups of nodes are scattered randomly in a square area at first, and a community is formed around the central position of each group. Then, nodes begin to walk inside their own community randomly. We assume that a node can walk beyond its current community to another one with a very small probability, P . The scenario is shown in Figure 2.

3.1.4. Problem Formulation. As shown in Figure 2, the relative positions among the vehicle nodes are stable since they show strong location and group preference. For example, for node 0, a great number of nodes walk on its north side and its south side, in which case it is unnecessary for node 0 to point its directional antenna to the east side and west side for a long time if it wants to discover more neighbor nodes in a short time. Moreover, as we mentioned, nodes can change their communities so that the scan pattern of node 0 can be out of date sometimes, in which case node 0 needs to adjust the scan pattern of its directional antenna to the updated topology around itself.

In order to successfully deliver the packets in an intermittent network, the source node has to send plenty of

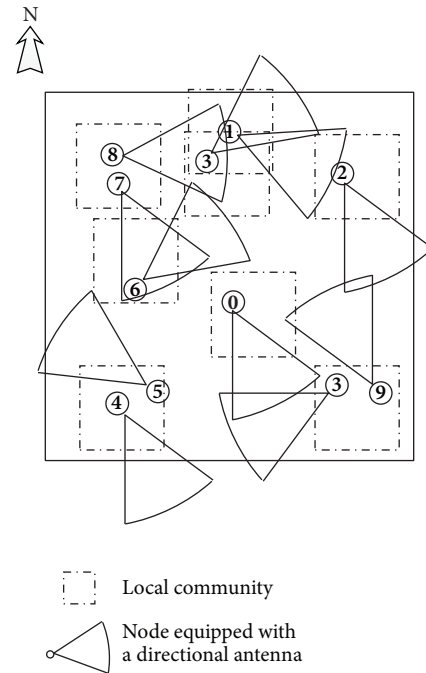


FIGURE 2: Mobility model.

data packet copies to the neighbors that it meets. However, it can cause heavy load of the network. Otherwise, if the source node simply chooses one of its neighbor nodes as the relay node, it cannot guarantee the successful delivery to the destination and leads to long delay. Therefore, the routing decision determines the transmission delay and the load and the energy consumption of the network.

For example, for node 8 in Figure 2, nodes 1, 3, 6, and 7 are the neighbor nodes maintained in its neighbor node list. When node 8 wants to send data packets to node 2, it needs to determine which neighbor nodes should be chosen to relay these packets. The answers of three questions about the neighbor nodes are critical when node 8 makes its routing decisions.

For a specific possible relay node:

- (1) What is the probability that node 8 can communicate with it?

- (2) If node 8 sends its data packets to this relay node, what is the probability that it can relay these data packets to node 2 in one hop?
- (3) What is the expected delay that it can successfully transmit these data packets to node 2?

Here, we can formulate the research objective of this paper:

To make the vehicle be able to adjust the scanning pattern of its directional antenna according to the topology around it and find out the optimal routing decisions to minimize the delay and the number of packet copies while successful packets delivery and short delay are guaranteed.

4. Neighbor Discovery

In this section, we devise a novel neighbor discovery algorithm with directional antennas which can make the nodes respond quickly to the topology changes. A priori knowledge is unnecessary when the nodes begin to detect their neighbor nodes.

To clearly demonstrate the protocol we devised, we assume that node j is within the cover range of node i and node k is within the cover range of node j while out of the cover range of node i .

4.1. Neighbor Node List. Neighbor node list is the fundamental part of our approach; sector distribution and routing decision are both based upon the information from the neighbor node list.

In our approach, nodes arrange their sectors according to the information in the neighbor node list. The neighbor node list of each node is built up and updated based on the interactive messages that it receives each time. The structure of the interactive message is shown in Figure 3:

- (1) ID: the ID of node i ;
- (2) Type: to indicate that the message is either a HELLO message or a response message;
- (3) Sector_S: to record the sector that the antenna of node i points to when node i broadcasts this message; it is denoted by S_i ($S_i \in \widehat{S}_i$);
- (4) TNS (time to next sector): to set the timer C_S , when it broadcasts this message.

The *Neighbor List* records all the neighbor nodes that node i has ever detected:

- (1) Neighbor ID: to indicate the ID of the neighbor node, such as the ID of node j ;
- (2) DES_N (duration for every sector of the neighbor node): to record all the sectors where node i has ever detected node j ; the total number of messages received from node j is denoted by $A_{ij}^{\widehat{S}_i}$;

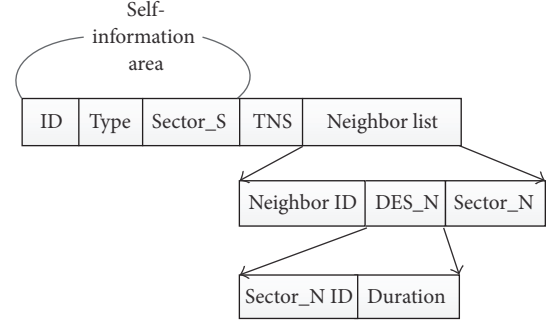


FIGURE 3: Structure of interactive message.

- (3) Sector_N (sector of the neighbor node for the last communication): to record the sector that the directional antenna of node i points to when node i detected node j last time, which is denoted by S_{ij} .

Every node maintains a neighbor node list. The format of each item in the neighbor node list is shown in Figure 4. The *Neighbor-Information Area* of node i records the information about node j . The *Two-Hop Node List* records all the information of the neighbor nodes that node j has ever detected, such as the information about node k . Their formats are the same as the HELLO message and their descriptions are similar as well, except that the *Time_Flag* indicates the update time of this item, which is updated based on the timer kept in node i since nodes are not time synchronized. It is denoted by T_{Flag}^i .

In order to improve the efficiency of neighbor discovery, nodes will send back a response message when they receive a HELLO message so that they can take every chance to make themselves discovered by others. However, collision issue is not concerned in this paper since nodes are not time synchronized like in [2–8]. We propose a mechanism to reduce the collision probability, which is that nodes are not forced to send back a response message every time when it receives a HELLO message. When a node, say j , receives a HELLO message from another node, say i , there are two steps to be taken:

- (1) Node j updates its neighbor node list according to the HELLO message from node i . The *Neighbor-Information Area* is updated according to the *Self-Information Area* of the HELLO message and the *Two-Hop Node List* is updated according to the *Neighbor List* of the HELLO message.
- (2) If node i is listed in the *neighbor node list* of node j and the information about itself is the latest, node j does not respond to node i . If not, node j sends back a response packet to node i after a random time period ε whether it is in the transmitting state or not. ε is supposed to be smaller than the *TNS* in the HELLO message from node i in case that node i turns its directional antenna to the next sector and it cannot receive the response message from node j .

As HELLO messages and response messages being broadcasted in the network, nodes are able to collect plenty of information to arrange their directional antennas.

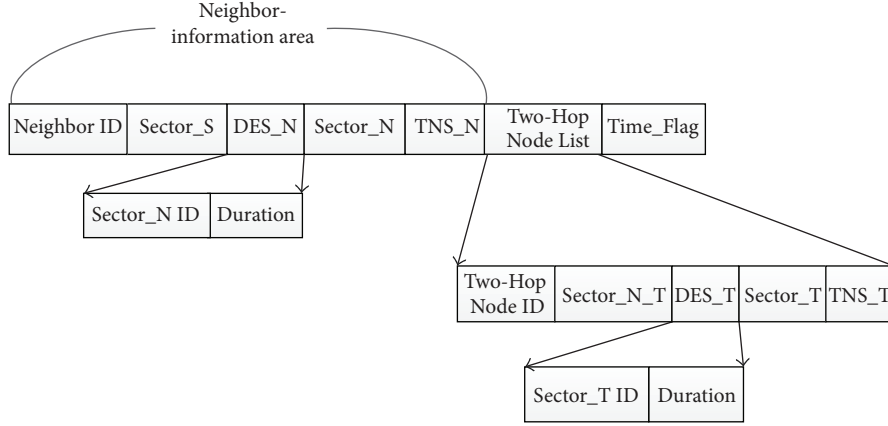


FIGURE 4: Neighbor node list.

4.2. Sector Steering Schedule Arrangement. In this part, we present how nodes adjust the steering time of their sectors based upon the neighbor node list.

As we mentioned in Section 3, when the mobility model of the nodes in the network shows strong location and group preference, nodes can waste time on the sectors where few neighbor nodes exist if they switch their directional antennas randomly. To make the nodes sensitive to the topology around them and arrange their directional antennas intelligently, we propose an algorithm called *History-Based Sector Distribution (HSD)*. At the beginning of *HSD*, node i switches its directional antenna to sectors randomly and the timer C_{S_i} ($S_i \in \widehat{S}_i$) is equal to $T\theta_i/2\pi$. As the neighbor discovery proceeds, node i will gather enough information to arrange its directional antennas and C_{S_i} ($S_i \in \widehat{S}_i$) is going to be adjusted. According to the DES_N in the neighbor node list maintained in node i , we can know which sectors communicate with other nodes frequently. They are the areas where groups of nodes may exist and are more likely for node i to discover new neighbor nodes.

Then, we can formulate the *HSD* as follows. When node i points its directional antenna to S_i , we can get that

$$R_{S_i} = \frac{\sum_{k \in L_i} A_{ik}^{S_i}}{\sum_{S \in \widehat{S}_i} \sum_{k \in L_i} A_{ik}^S}, \quad (1)$$

where R_{S_i} represents the proportion of interactive messages received through S_i compared to the total messages received through all sectors of node i and L_i represents the set of neighbor nodes recorded in the neighbor node list of node i . We define $Q_{S_i} \triangleq \sum_{k \in L_i} A_{ik}^{S_i}$ and $E_i \triangleq \sum_{S \in \widehat{S}_i} \sum_{k \in L_i} A_{ik}^S$. The relationship between R_{S_i} and C_{S_i} is expressed as follows:

$$C_{S_i} = R_{S_i} T = \frac{Q_{S_i}}{E_i} T. \quad (2)$$

In the preliminary stage, Q_{S_i} and E_i in (2) are both very small and the sector which is the first to discover neighbor

nodes tends to have a bigger R_{S_i} since Q_{S_i} and E_i can be all generated by this sector, in which case the node possibly points its antenna to this sector for an inappropriate C_{S_i} which is close to T . With the process of neighbor discovery, according to (1) and (2), C_{S_i} of each sector would be too stable to make a response to the topology changes. For example, C_{S_i} of the sectors where few neighbor nodes are discovered tend to be small or even close to zero. However, nodes can change their communities with a small probability as we mentioned in Section 3. Moreover, the relative positions among nodes can change dramatically sometime even if they walk in the same community, such as when nodes walk to an opposite direction. Therefore, neighbor nodes can appear in the sectors which used to be empty and the previous C_{S_i} of these sectors would be obsolete.

To make (2) sensitive to the topology of the network, in *HSD*, we propose a mechanism to estimate the topology changes by monitoring the frequency of message receiving.

Firstly, in a specific monitoring time period, M_i , then, we can get

$$V_{S_i} = \frac{Q_{S_i}}{M}, \quad (3)$$

where V_{S_i} indicates the message receiving frequency of S_i in the recent time period of M_i . Obviously, V_{S_i} is a stable value when the relative positions among the nodes are stable. In that case, the instant receiving frequency, V'_{S_i} , is stable as well and it holds that

$$V'_{S_i} = \frac{\Delta Q_{S_i}}{\Delta t} \cong V_{S_i}. \quad (4)$$

According to (1) and (3), we notice that

$$\frac{V_{S_i}}{R_{S_i}} = \frac{E_i}{M_i}. \quad (5)$$

Therefore, V_{S_i}/R_{S_i} is only determined by the status of node i but not the status of S_i . Due to this, we define $\varphi_i \triangleq E_i/M_i$

which is the message receiving frequency of node i regardless of the status of its sectors.

According to (4) and (5), when the topology in the network is stable, the following is established between every two sectors of node i :

$$\frac{V'_{S_i^m}}{R_{S_i^m}} \cong \frac{V'_{S_i^n}}{R_{S_i^n}} \cong \frac{E_i}{M} \quad (S_i^m, S_i^n \in \widehat{S}_i; m \neq n). \quad (6)$$

Therefore, we can compare V'_{S_i} among \widehat{S}_i to estimate the topology variation around node i based on (6). The functioning process of (6) is analyzed as follows.

In the preliminary stage of the neighbor discovery process, according to (6), the adjustment of R_{S_i} would be started after the first monitoring time period. And node i is supposed to scan more than two sectors during the first monitoring time period to apply (6). Hence, the initial C_{S_i} would be restricted within a reasonable range due to the comparison of C_{S_i} among sectors. After that, when the topology around the node i changes, there are two situations:

- (i) The relative positions among nodes are changed while the neighbor nodes are still in the transmission range of node i ; that is, the number of neighbor nodes within the transmission range of node i remains the same;
- (ii) Some neighbor nodes are no longer in the transmission range of node i or new neighbor nodes come into the transmission range; that is, the number of neighbor nodes within the transmission range of node i is changed.

The simulation result of [1] shows that in some cases nodes cannot discover each other even though they are within the transmission range of each other due to the equipment of directional antenna, while, in general, more neighbor nodes within the transmission range of node i can lead to more frequent communications. Thus, we assume that φ_i is only determined by the number of nodes within the transmission range of node i .

For the first situation, for example, some nodes move from sector S_i^m to sector S_i^n . Obviously, $V'_{S_i^m}$ will decrease while $V'_{S_i^n}$ will increase. Then we can readjust $R_{S_i^m}$ and $R_{S_i^n}$ according to (6) since φ_i is stable.

For the second situation, φ_i is going to change as well, and we need to rewrite (6) to the following form:

$$R_{S_i} \cong \frac{V'_{S_i}}{\varphi_i}. \quad (7)$$

When the second situation happens, to clearly figure out the variation of R_{S_i} , we separately analyze the variation of V'_{S_i} and φ_i as follows:

$$\begin{aligned} \Delta V'_{S_i} &= \frac{\Delta Q_{S_i}}{\Delta t} - (V'_{S_i})_{\text{before}}, \\ \Delta \varphi_i &= \frac{E_i + \Delta E_i}{M + \Delta t} - (\varphi_i)_{\text{before}}; \end{aligned} \quad (8)$$

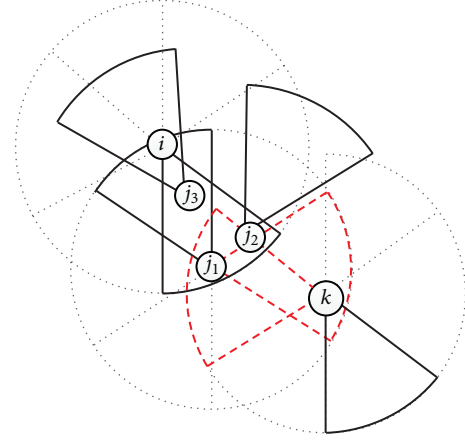


FIGURE 5: Relay node selection.

then,

$$\begin{aligned} \Delta V'_{S_i} - \Delta \varphi_i &= \frac{\Delta Q_{S_i}}{\Delta t} - \frac{E_i + \Delta E_i}{M + \Delta t} + (\varphi_i)_{\text{before}} \\ &\quad - (V'_{S_i})_{\text{before}}. \end{aligned} \quad (9)$$

Simplifying (9) and letting

$$\begin{aligned} \frac{\Delta Q_{S_i} - \Delta E_i}{M} &\rightarrow 0, \\ \frac{\Delta t}{M} &\rightarrow 0 \end{aligned} \quad (10)$$

yield

$$\Delta V'_{S_i} - \Delta \varphi_i \cong V'_{S_i} - (V'_{S_i})_{\text{before}}. \quad (11)$$

According to (11), when new neighbor nodes come into the transmission range of node i such as appearing in sector S_i , it is obvious that V'_{S_i} would be bigger than $(V'_{S_i})_{\text{before}}$ so that $\Delta V'_{S_i} > \Delta \varphi_i$. According to (7), R_{S_i} would increase. Contrarily, when nodes flee from S_i , V'_{S_i} would be smaller than $(V'_{S_i})_{\text{before}}$ so that R_{S_i} would decrease. By far we can conclude that (6) works well for dynamic topology.

5. Routing Protocol

In the network where nodes are equipped with directional antennas, there is no consistent path from the source node to the destination node. In this section, we devise a special routing protocol called *HSD-R* which is formulated based upon HSD. The routing decisions are made based on the knowledge maintained in the neighbor node list.

5.1. Neighbor Node Table Processing. We take Figure 5 as an example to demonstrate how we analyze the neighbor node table when there are data packets waiting to be transmitted.

As illustrated in Figure 5, when node i plans to transmit its data packets to node k , it can send its packets to nodes j_1 and

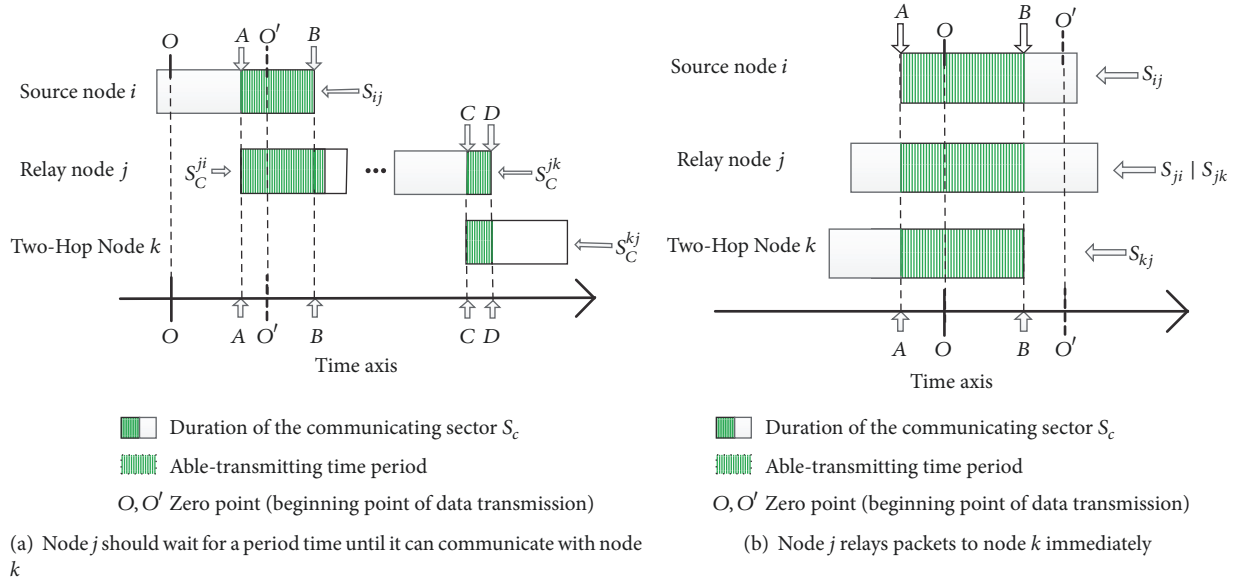


FIGURE 6

j_3 immediately. But node j_3 is not qualified to be a relay node since node k is beyond its transmission distance. However, for node j_1 , we need to take a further consideration because the connecting time between it and node i could be too short to finish a complete data transmission when they point their directional antennas to each other. Moreover, when data packets arrive at node j_1 , the time consumption for node j_1 and node k to point their directional antennas to a proper direction also needs to be considered since the data packets are conserved in node j_1 during that time and cannot be transmitted.

In order to make an appropriate routing decision, we need to find out which nodes can relay the data packets to the destination successfully with a high probability. When node i wants to communicate with node j through sector S_i , the communicating probability can be expressed as follows:

$$P_{ij}^{S_i} = \frac{A_{ij}^{S_i}}{(M/(n+1)\tau)R_{S_i}}, \quad (12)$$

where $P_{ij}^{S_i}$ refers to the communication probability between node i and node j through S_i . Furthermore, we need to make sure that S_i is the best option for node i to communicate with node j . For example, in the most recent monitoring time period, node i and node j have a frequent communication in S_i but this kind of situation could be a result of that node i and node j just happen to detect each in the recent M while they seldom encountered with each other before. In other words, the neighbor relationship between node i and node j existing in S_i is only during the most recent monitoring time period. For this kind of neighbor nodes, they are not good options for relaying data packets due to the weak neighbor relationship between them and the source node. Therefore, we introduce a coefficient, σ_{ij} , which refers to the stability of neighborhood

relationship between node i and node j . It can be calculated as follows:

$$\sigma_{ij} = \frac{\sum_{S \in \bar{S}_i} A_{ij}^S}{E_i}. \quad (13)$$

Before analyzing the time metric of every possible path, we need to detail how the directional antennas operate during the data transmission. As we mentioned in Section 3, nodes are supposed to switch their directional antennas in a predictable sequence in order to make the routing decisions more precise, while the source node can begin its data transmission at any time point and nodes are not time synchronized in this paper. Therefore, we propose a mechanism to make nodes to change their rotation pattern spontaneously. When there are data packets conserved in the nodes and waiting to be transmitted, data packets could be generated by the nodes themselves or received from other neighbors, these nodes switch their rotation mode to $DT-M$. Otherwise, they still keep their scan pattern as $PTN-M$. The broadcast of interactive messages is still going on whether the rotation mode is $DT-M$ or $PTN-M$ since the topology changes can also happen during the data transmissions.

Considering that nodes make their routing decisions based on the status of nodes in two-hop distance, there are at least three sets of directional antennas participating in data transmission. According to the neighbor node list maintained in the source node i , we can construct the relationships about sectors among the source node, the neighbor nodes, and the two-hop nodes. As illuminated in Figure 6, we can draw the position and length of the frame for *Source Node i* according to the *Sector Distribution* in neighbor node list; the position and length of the frames for *Relay Node j* and *Two-Hop Node k* are separately decided by the *Neighbor-Information Area* and the *Two-Hop Node List*. In Figure 6, the zero point of the

time axis, T_0 , refers to the time point when the source node prepares to send its data packets. The green zone between point A and point B or between point C and point D refers to the time period when the directional antennas of two nodes cover each other; that is, the two nodes can communicate with each other during the green zone.

When the relay node receives data packets successfully, there are two possibilities: (1) the relay node waits for a period of time until it can communicate with node k , such as shown in Figure 6(a); (2) the relay node sends them to node k immediately, which means $S_{jk} = S_{kj}$, as shown in Figure 6(b).

Source node, like node i , can send data packets at any time. In other words, the position of point O can be anywhere within the duration of S_{ij} , while, in general, it can be categorized into two cases: (1) it is ahead of the able-transmitting period, like point O in Figure 6(a); (2) it is during the able-transmitting period, like point O' in Figure 6(a) and point O in Figure 6(b). If the node misses the two kinds of time positions, like point O' in Figure 6(b), it has to wait for the next communicating sector and it can be considered as point O again.

From Figure 6 we can see that, in order to figure out the relationship among the three sets of directional antennas, it is necessary to figure out the beginning time points, T_{begin}^{mn} ($m, n \in \{i, j, k\}; m \neq n$), and the ending time points, T_{end}^{mn} ($m, n \in \{i, j, k\}; m \neq n$), for these sectors:

(1) S_{ji} :

$$\begin{aligned} T_{\text{end}}^{ji} &= T_{\text{Flag}}^j + C_{S_{ji}} \\ T_{\text{begin}}^{ji} &= T_{\text{end}}^{ji} - R_{S_{ji}} T. \end{aligned} \quad (14)$$

(2) S_{jk} :

$$T_{\text{end}}^{jk} = \begin{cases} T_{\text{end}}^{ji}, & S_{ji} = S_{jk} \\ T_{\text{end}}^{ji} + \sum_{m=1+S_{ji}}^{S_{jk}} R_m T, & S_{ji} \neq S_{jk} \end{cases} \quad (15)$$

$$T_{\text{begin}}^{jk} = T_{\text{end}}^{jk} - R_{S_{jk}} T.$$

(3) S_{kj} :

$$\begin{aligned} T_{\text{end}}^{kj} &= T_{\text{Flag}}^k + C_{S_{jk}} \\ T_{\text{begin}}^{kj} &= T_{\text{end}}^{kj} - R_{S_{kj}} T. \end{aligned} \quad (16)$$

Then we present an algorithm to predict the waiting and transmitting time for every possible path.

After the waiting time and able-transmitting time of each path are calculated according to Algorithm 1, the source node is able to make its routing decisions.

5.2. Routing Decisions. When the source node, say node i , wants to send its data packets to the destination node, say node k , we assume that the set of the neighbor nodes discovered by node i is denoted by L_i and there are β_m ($m \in L_i$) two-hop nodes in the *Two-Hop Node List* of each neighbor node.

Therefore, the number of potential two-hop paths for the node i is $\sum_{m \in L_i} \beta_m$. And there are four possibilities when node i goes through its neighbor node table:

- (a) Node k is only recorded in the *Neighbor-Information Area*.
- (b) Node k is only recorded in the *Two-Hop Node List* of some neighbor nodes.
- (c) Node k is recorded not only in the *Neighbor-Information Area* but also in the *Two-Hop Node Lists*. In other words, node i can send its data packets to node k directly or via the relay nodes.
- (d) There is no record for node k in the neighbor node list.

For the first three possibilities, every possible path should be taken into a further consideration because direct paths may have less connecting time or lower connecting ratio. The set of paths that can reach the destination node is denoted by \bar{U} .

From Algorithm 1 we can see that the time consumption for a complete transmission only consists of the able-transmitting time and the waiting time. For a specific path from the source node to the destination node, obviously, the decisive part for a successful delivery is the weakest link among the paths, which refers to the most unstable neighbor relationship between two nodes. The calculation results of Algorithm 1 are theoretical time based upon the most recently updated history records. To make it more suitable for a general situation, we propose an inequation to filter every path in \bar{U} :

$$\min \left\{ P_{ij}^{S_{ij}} T_{\text{trans1}}^h, P_{jk}^{S_{jk}} T_{\text{trans2}}^h \right\} > H, \quad (17)$$

where $P_{ij}^{S_{ij}} T_{\text{trans1}}^h$ and $P_{jk}^{S_{jk}} T_{\text{trans2}}^h$ indicate the predicted time period for delivery of each hop and H indicates the sum of transmission delay and propagation delay for one hop. In this paper, the propagation delay between the nodes is assumed to be small and the size of data packets is the same; therefore, we hypothesize that H is the same for every node. Equation (17) indicates that the minimum of the able-transmitting time period of an ideal path should be enough to complete one packet transmission.

In order to reduce the load of the network, we set a TTL, \mathbb{T} , for every data packet. Based upon the previous discussion, we can conclude that the TTL for the data packets in a specific path should be the sum of H and the waiting time in the path. To be more tolerant of the complex environment condition, we set \mathbb{T} as follows:

$$\mathbb{T} = \max_{m \in \bar{U}} T_{\text{waiting}}^m + H. \quad (18)$$

Algorithm 2 details the procedure for the path selection. The set of qualified paths is denoted by \bar{V} .

After the path selection process, the source node generates ν copies of its data packets and transmits them through the first ν paths in \bar{V} . ν is a preset constant number which is detailed in Section 6.


```

Input:  $T_O, T_{begin}^{ij}, T_{end}^{ij}, T_{Flag}^j, T_{Flag}^k$ 
Output:  $T_{waiting}^h, T_{trans1}^h, T_{trans2}^h$ 
for every row of the neighbor node table do
  Calculate the beginning and ending time points for  $S_{ji}, S_{jk}$  (Eq. (14)-(15)).
for every row of the Two-hop Node List do
  Calculate the beginning and ending time points for  $S_{kj}$  (Eq. (16)).
  if  $S_{ji} = S_{jk}, T_{end}^{ij} > T_{begin}^{kj}$  then
    if  $T_O < \max\{T_{begin}^{ij}, T_{begin}^{ji}, T_{begin}^{kj}\}$  then
      Set  $T_{waiting}^h = \max\{T_{begin}^{ij}, T_{begin}^{ji}, T_{begin}^{kj}\} - T_O$ 
      Set  $T_{trans1}^h = T_{trans2}^h = \min\{T_{end}^{ij}, T_{end}^{ji}, T_{end}^{kj}\} - \max\{T_{begin}^{ij}, T_{begin}^{ji}, T_{begin}^{kj}\}$ 
    else then
      Set  $T_{waiting}^h = 0$ 
      Set  $T_{trans1}^h = T_{trans2}^h = \min\{T_{end}^{ij}, T_{end}^{ji}, T_{end}^{kj}\} - T_O$ 
    end if
  else then
    Identify  $T_{waiting}^h$  as the waiting time for the first hop of path  $h$ 
    Identify  $T_{waiting}^{\prime h}$  as the waiting time for the second hop of path  $h$ 
    if  $T_O < \max\{T_{begin}^{ij}, T_{begin}^{ji}\}$  then
       $T_{waiting}^h = \max\{T_{begin}^{ij}, T_{begin}^{ji}\} - T_O$ 
      Set  $T_{trans1}^h = \min\{T_{end}^{ij}, T_{end}^{ji}\} - \max\{T_{begin}^{ij}, T_{begin}^{ji}\}$ 
    else then
       $T_{waiting}^h = 0$ 
      Set  $T_{trans1}^h = \min\{T_{end}^{ij}, T_{end}^{ji}\} - T_O$ 
    end if
    Set  $T_{waiting}^{\prime h} = \max\{T_{begin}^{jk}, T_{begin}^{kj}\} - \min\{T_{end}^{ij}, T_{end}^{ji}\}$ 
    Set  $T_{waiting}^h = T_{waiting}^h + T_{waiting}^{\prime h}$ 
    Set  $T_{trans2}^h = \min\{T_{end}^{jk}, T_{end}^{kj}\} - \max\{T_{begin}^{ij}, T_{begin}^{ji}\}$ 
  end if
end for
end for

```

ALGORITHM 1: Calculation for the waiting and transmitting time.

```

Input:  $\tilde{U}, S_{ij}, k$ 
Output:  $\tilde{V}$ 
for every path  $\tilde{u} \in \tilde{U}$  do
  Identify Sector  $S$  in  $\tilde{u}$  as  $S_s$ 
  Identify  $\omega^h = \min\{P_{ij}^{S_{ij}} T_{trans1}^h, P_{jk}^{S_{jk}} T_{trans2}^h\}$ 
  if  $S_i = S_s$  and  $\omega^h > H$  then
     $\tilde{V} \leftarrow \tilde{V} + \{\tilde{u}\}$ 
  end if
end for
if  $\tilde{V}$  is null then
  The source node holds its data packets and continue the neighbor discovery process until it finds qualified paths.
end if
Sort  $\tilde{V}$  in descending order according to  $\sigma_{ij} \sigma_{jk} \omega^h$ 

```

ALGORITHM 2: Path selection.

```

Input:  $x$ 
Output:  $\tilde{N}$ 
  for every neighbor node  $j \in L_i$  do
    if  $\beta_j > x$  then
       $\tilde{N} \leftarrow \tilde{N} + \{j\}$ 
    end if
  end for
  if  $\tilde{N}$  is null then
    for every neighbor node  $j \in L_i$  do
      if  $j$  has different neighbor nodes compared to  $L_i$  then
         $\tilde{N} \leftarrow \tilde{N} + \{j\}$ 
      end if
    end for
  end if

```

ALGORITHM 3: Extreme situation treatment.

Then, we discuss the fourth situation that there is no record for node k in the neighbor node list. Apparently, a node that has encountered more neighbor nodes may get more topology information in its two-hop distance. Hence, when this kind of extreme situation occurs, node i sends its data packets to all the neighbor nodes who have discovered more neighbor nodes than node i dose; that is, node i sends its data packets to node j if β_j is bigger than the size of L_i . If there is still no qualified neighbor node after that, then we choose those nodes who have encountered different neighbor nodes compared to L_i . Algorithm 3 describes this procedure. The size of L_i is denoted by x , and the set of qualified neighbor nodes is denoted by \tilde{N} .

6. Simulation Results

In this section, we evaluate the performance of *HSD* and *HSD-R* separately. We use a simulation setup that consists of a uniform distribution of nodes over a 2-dimensional plane of area $20 \text{ km} \times 20 \text{ km}$. The fixed transmission range is $R = 10 \text{ km}$ and the angle of each directional antenna is $\pi/3$, that is, 6 sectors for each vehicle node. The rotation period of the directional antennas is $T = 180\tau$ and the duration of each receiving state is $n = 9$ which indicates that each node can broadcast HELLO messages 18 times in a circle totally. The monitoring time $M = 360\tau$. The moving speed of the vehicle nodes is 50 km/h which is nearly 15 meters every 180τ with a random direction.

We vary the total number of nodes inside this area with 5, 10, and 50, and they represent three levels of node density which are extremely low, low, and high, respectively. We assume that the area of the community is $4 \text{ km} \times 4 \text{ km}$, and the number of nodes in one community is random which is from 2 to 5 and $P = 0.1$. In the extremely low density scenario, we artificially divide the 5 nodes into two groups with 2 and 3 separately. We run the simulation 20 times and the simulation duration is $10^4\tau$. Nodes are randomly scattered inside the area every time. There is a chance that nodes cannot discover all other nodes because their directional antennas cannot cover the whole area.

6.1. Evaluation of *HSD*. Firstly, we compare and analyze the performance of *HSD* and the random algorithm. When the nodes switch their sectors clockwise with an equal and fixed C_S , it turns out that their average time consumption is much larger than *HSD* and the random algorithm. Therefore, we only concentrate on the simulation comparison of *HSD* and the random algorithm. Figure 7 shows the performances of the two algorithms on every node to successfully discover a certain percentage of neighbor nodes inside the area.

In Figure 7(c), *HSD* and the random algorithm show no obvious difference on time consumption when the discovered nodes are under 70% since, in the high density scenario, there are plenty of neighbor nodes in any sectors so that the scan pattern of *HSD* is similar to the random algorithm. When the discovered nodes reach 70%, the performance of *HSD* is worse. Because higher density of nodes leads to more frequent communication among nodes, the nodes especially which are located at the central part of the scenario can go through turbulence on the schedule of sectors. However, as shown in Figures 7(a) and 7(b), the *HSD* achieves better performance since the probability of communication is smaller and the change of topology is easier to be detected when the density of nodes is low. Especially in Figure 7(a), when nodes in one community have discovered a node in another community, they can point their directional antennas to the certain sector where another community locates for a long time since there are few interferences from other sectors. Hence, the average time consumption and the standard deviation of time consumption of *HSD* are both smaller than those of the random algorithm. According to [20], the packet size has obvious influence on the performance of protocols. However, the interactive action among nodes is rare since the node density in this paper is quite low; for example, only one neighbor node exists within a sector for the low density scenario.

In summary, simulation results demonstrate that *HSD* achieves better performance than the random algorithm on neighbor discovery when nodes are sparsely scattered.

6.2. Evaluation of *HSD-R*. From the simulation results of *HSD*, we conclude that *HSD* achieves better performance

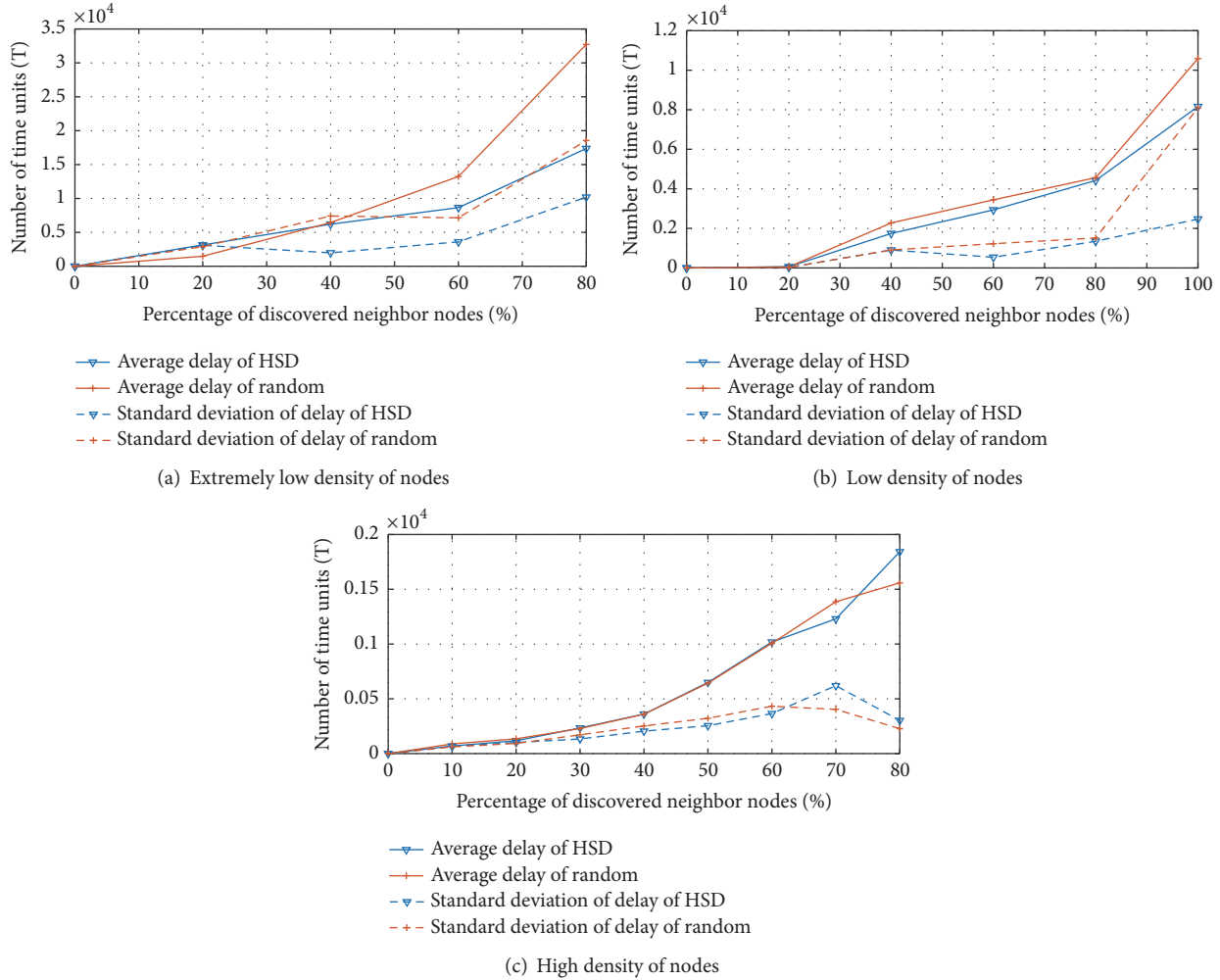


FIGURE 7: Simulation result comparison on different node densities.

when nodes are sparsely scattered. Reference [16] proposes a routing protocol called *Spray and Focus*. Though *Spray and Focus* is not designed for the networks where directional antennas are equipped, it is a classic routing protocols for opportunistic networks and the most important is that nodes in [16] show location and group preference as well. Therefore, we compare and analyze the performance of *HSD-R* and *Spray and Focus* in low density scenario and high density scenario separately. *Spray and Focus* consists of two phases: in the first phase it distributes a fixed number of copies to the first few relays encountered, and in the second phase each relay can forward its copy to a potentially more appropriate relay, using a carefully designed utility-based scheme.

According to [16], the number of packet copies equal to about 5–10% of the total nodes serves as a useful rule of thumb for good performance. Because nodes in [16] are not equipped with the directional antennas, the connecting probability among nodes in our scenario is much lower than that in [16]. Under this consideration, we increase the number of packets copies, that is, the number of selected paths, ν , gradually from approximately 5% to 20% of the total nodes to compare their performance. The beginning time of data

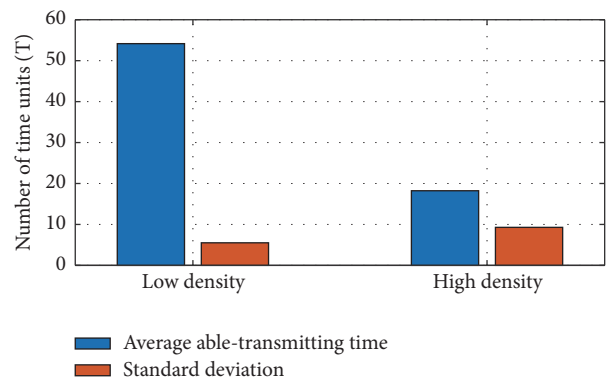


FIGURE 8: Average able-transmitting time in low density scenario and high density scenario with ν equal to 25% of the total nodes.

transmission for every node is random and the destination for every source node is selected randomly as well.

When the node density is low, only two paths can be selected from \tilde{V} and around ten paths for high density scenario. As illuminated in Figure 8, the average able-transmitting

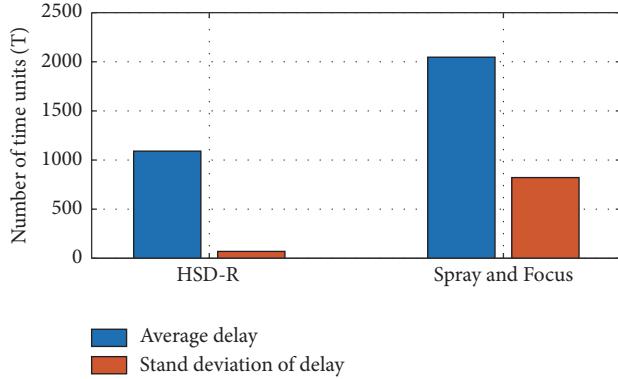
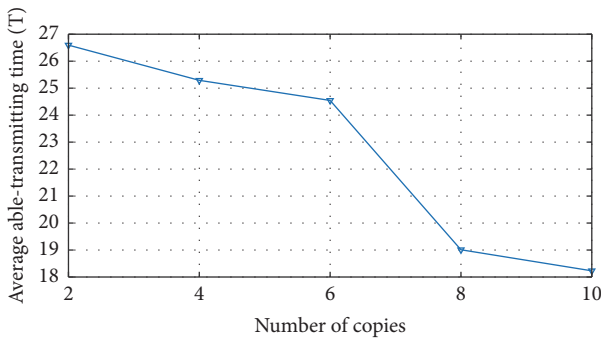


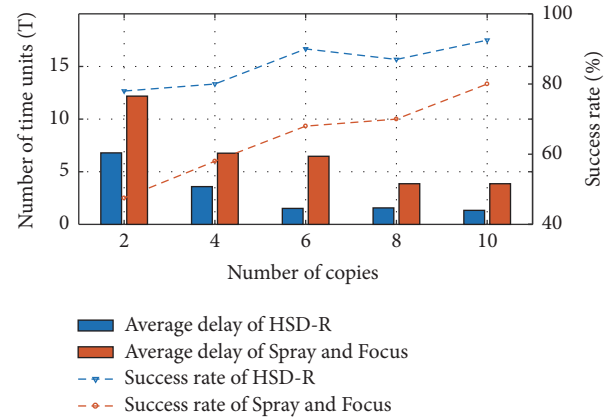
FIGURE 9: Average delay in low density scenario.

FIGURE 10: Average able-transmitting time in high density scenario with different v .

time, T_{trans}^h , among nodes in low density scenario is bigger since nodes mostly point their directional antennas to certain sectors where another community locates for a long time and there are few interferences from other sectors.

Figure 9 depicts the performance of the two protocols in low density scenario. Compared to *Spray and Focus*, the transmission delay of *HSD-R* is much lower and it shows no obvious differences for every node. Figure 9 indicates that the selected paths in *HSD-R* do reduce the delay. The successful transmission rates of the two protocols in low density scenario are all above 90%.

When the node density gets higher, in Figure 10, we can see an obvious decrease on T_{trans}^h when the number of copies is bigger than 6, that is, around 10% of the total nodes. It indicates that the average able-transmitting time of the first 6 paths is much bigger than the left paths. Figure 11 proves that the minimum number of copies for *HSD-R* to achieve a reasonable success rate and delay is around 10% as well. Moreover, we can see that the success rate of *Spray and Focus* decreases dramatically when the number of copies is small while the transmission delay of *HSD-R* is slightly changed. And the minimum of number copies for *HSD-R* to achieve a reasonable performance is smaller than *Spray and Focus*. What needs to be explained is that the obvious success rate difference between *HSD-R* and *Spray and Focus* is caused by small number of total nodes in the scenario, and few data packets failing to be delivered to the destination can lead to a great decrease in success rate. We do not have the simulation

FIGURE 11: Success rate and average delay in high density scenario with different v .

on extremely high density scenarios because the directional antenna is not designed for that kind of scenarios.

In summary, compared to *Spray and Focus*, *HSD-R* can guarantee a successful delivery with a smaller number of packet copies while directional antennas are equipped.

7. Conclusion

In this paper, we concentrate on the networking of vehicles. Firstly, we have proposed a novel neighbor discovery algorithm called *History-Based Sector Distribution (HSD)* for vehicle ad hoc networks where directional antenna is equipped. The proposed *HSD* can make the vehicles be able to arrange their directional antennas according to the topology of the network. We have proved that our algorithm can achieve better performances than the traditional random algorithm especially in the scenario where vehicle nodes are sparsely scattered. Secondly, we design a routing protocol called *HSD-R* which is derived from *HSD*. In *HSD-R*, vehicle nodes make their routing decisions by analyzing the link quality of each possible path between the source node and the destination node based on the information collected during the neighbor discovery process. The evaluation results of *HSD-R* prove that high successful transmission rate and low transmission delay are guaranteed with a small number of packet copies.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported in part by the National Key R&D Plan of China (Grants 2016YFB1200100).

References

- [1] L. Chen, Y. Li, and A. V. Vasilakos, "Oblivious neighbor discovery for wireless devices with directional antennas," in

- Proceedings of the 35th Annual IEEE International Conference on Computer Communications, IEEE INFOCOM 2016*, pp. 1–9, April 2016.
- [2] S. Huang, M. Li, and L. Zhao, “An intelligent neighbor discovery algorithm for ad hoc networks with directional antennas,” in *Proceedings of the 2013 International Conference on Mechatronic Sciences, Electric Engineering and Computer, MEC 2013*, pp. 302–305, chn, December 2013.
 - [3] W. Xiong, B. Liu, and L. Gui, “Neighbor discovery with directional antennas in mobile ad-hoc networks,” in *Proceedings of the 54th Annual IEEE Global Telecommunications Conference: “Energizing Global Communications”, GLOBECOM 2011*, USA, December 2011.
 - [4] S. Zhao, Y. Liu, T. Yang, Z. Feng, Q. Zhang, and C. Gao, “3-Way multi-carrier asynchronous neighbor discovery algorithm using directional antennas,” in *Proceedings of the 2016 IEEE Wireless Communications and Networking Conference, WCNC 2016*, Qatar, April 2016.
 - [5] H. Cai and T. Wolf, “On 2-way neighbor discovery in wireless networks with directional antennas,” in *Proceedings of the 34th IEEE Annual Conference on Computer Communications and Networks, IEEE INFOCOM 2015*, pp. 702–710, Hong Kong, May 2015.
 - [6] H. Cai, B. Liu, L. Gui, and M.-Y. Wu, “Neighbor discovery algorithms in wireless networks using directional antennas,” in *Proceedings of the 2012 IEEE International Conference on Communications, ICC 2012*, pp. 767–772, Canada, June 2012.
 - [7] Z. Zhang and B. Li, “Neighbor discovery in mobile ad hoc self-configuring networks with directional antennas: Algorithms and comparisons,” *IEEE Transactions on Wireless Communications*, vol. 7, no. 5, pp. 1540–1549, 2008.
 - [8] Z. H. Mir, W.-S. Jung, and Y.-B. Ko, “Continuous neighbor discovery protocol in wireless ad hoc networks with sectored-antennas,” in *Proceedings of the 29th IEEE International Conference on Advanced Information Networking and Applications, AINA 2015*, pp. 54–61, Republic of Korea, March 2015.
 - [9] F. Tian, R. Q. Hu, Y. Qian, B. Rong, B. Liu, and L. Gui, “Pure asynchronous neighbor discovery algorithms in ad hoc networks using directional antennas,” in *Proceedings of the 2013 IEEE Global Communications Conference, GLOBECOM 2013*, pp. 498–503, USA, December 2013.
 - [10] F. Tian, B. Liu, H. Cai, H. Zhou, and L. Gui, “Practical Asynchronous Neighbor Discovery in Ad Hoc Networks with Directional Antennas,” *IEEE Transactions on Vehicular Technology*, vol. 65, no. 5, pp. 3614–3627, 2016.
 - [11] E. Gelai, G. Jakllari, S. V. Krishnamurthy, and N. E. Young, “An integrated scheme for fully-directional neighbor discovery and topology management in mobile ad hoc networks,” in *Proceedings of the 2006 IEEE International Conference on Mobile Ad Hoc and Sensor Systems, MASS*, pp. 139–149, Canada, October 2006.
 - [12] Z. Huang, C.-C. Shen, C. Srisathapornphat, and C. Jaikaeo, “Topology control for ad hoc networks with directional antennas,” in *Proceedings of the 11th International Conference on Computer Communications and Networks, ICCCN 2002*, pp. 16–21, USA, October 2002.
 - [13] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, “Efficient routing in intermittently connected mobile networks: the single-copy case,” *IEEE/ACM Transactions on Networking*, vol. 16, no. 1, pp. 63–76, 2008.
 - [14] V. Conan, J. Leguay, and T. Friedman, “Fixed point opportunistic routing in delay tolerant networks,” *IEEE Journal on Selected Areas in Communications*, vol. 26, no. 5, pp. 773–782, 2008.
 - [15] S. Shaghaghian and M. Coates, “Optimal forwarding in opportunistic delay tolerant networks with meeting rate estimations,” *IEEE Transactions on Signal and Information Processing over Networks*, vol. 1, no. 2, pp. 104–116, 2015.
 - [16] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, “Spray and focus: efficient mobility-assisted routing for heterogeneous and correlated mobility,” in *Proceedings of the 5th Annual IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops ’07)*, pp. 79–85, March 2007.
 - [17] Z. Chen and C. Chen, “Self-stopping epidemic routing in cooperative wireless mobile sensor networks,” in *Proceedings of the 12th IEEE International Conference on Wireless and Mobile Computing, Networking and Communications, WiMob 2016*, USA, October 2016.
 - [18] C. Y. Aung, I. W.-H. Ho, and P. H. J. Chong, “Store-Carry-Cooperative Forward Routing with Information Epidemics Control for Data Delivery in Opportunistic Networks,” *IEEE Access*, vol. 5, pp. 6608–6625, 2017.
 - [19] C. Wu, S. Ohzahata, Y. Ji, and T. Kato, “How to Utilize Interflow Network Coding in VANETs: A Backbone-Based Approach,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 8, pp. 2223–2237, 2016.
 - [20] C. Wu, X. Chen, Y. Ji et al., “Packet size-aware broadcasting in VANETs with fuzzy logic and RL-based parameter adaptation,” *IEEE Access*, vol. 3, pp. 2481–2491, 2015.
 - [21] T. Noguchi and T. Kobayashi, “Adaptive Location-Aware Routing with Directional Antennas in mobile adhoc networks,” in *Proceedings of the 2017 International Conference on Computing, Networking and Communications, ICNC 2017*, pp. 1006–1011, USA, January 2017.
 - [22] J. Yoon, W. Shin, and S. Jeon, “Elastic Routing in Ad Hoc Networks with Directional Antennas,” *IEEE Transactions on Mobile Computing*, vol. 16, no. 12, pp. 3334–3346, 2017.



Hindawi

Submit your manuscripts at
www.hindawi.com

