Paper

LDAOR – Location and Direction Aware Opportunistic Routing in Vehicular Ad hoc Networks

Marziyeh Barootkar[1], Akbar Ghaffarpour Rahbar*[1], and Masoud Sabaei[2]

[1] *Computer Networks Research Lab, Electrical Engineering Technologies Research Center, Sahand University of Technology Tabriz, Iran*
[2] *Computer Engineering and Information Technology Department, Amirkabir University of Technology Tehran, Iran*
* *Corresponding Author*

Abstract—Routing in Vehicular Ad hoc Networks (VANETs) has found significant attention because of its unique features such as lack of energy constraints and high-speed vehicles applications. Besides, since these networks are highly dynamic, design process of routing algorithms suitable for an urban environment is extremely challenging. Appropriate algorithms could be opportunistic routing (OR) where traffic transmission is performed using the store-carry-forward mechanism. An efficient OR mechanism, called Location and Direction Aware Opportunistic Routing (LDAOR), is proposed in this paper. It is based on the best neighbor node selection by using vehicles positions, vehicles directions, and prioritization of messages from buffers, based on contact histories and positions of neighbor nodes to destination. In LDAOR, when multiple nodes make contact with a carrier node, the closest neighbor node to destination is selected as the best forwarder. However, when only one node makes contact with the carrier node, the message is delivered to it if it moves toward the destination. Using the ONE simulator, the obtained performance evaluation results show that the LDAOR operates better than conventional OR algorithms. The LDAOR not only increases delivery rate, but also reduces network overhead, traffic loss, and number of aborted messages.

Keywords—carry-and-forward mechanism, contact history knowledge, direction and location aware routing, opportunistic routing, vehicular ad hoc networks.

1. Introduction

The routing algorithms designed for Mobile Ad Hoc Networks could not be appropriate for Vehicular Ad Hoc Networks (VANETs) since they do not consider inherent features of VANET such as high mobility of vehicles (that leads to frequent topology changes and unstable links), and short-time connections among vehicles [1], [2]. Therefore, new routing algorithms must be designed in such a way that no packet is lost when connections are disconnected. To solve this problem, opportunistic routing (OR) algorithms have been proposed for VANETs [3], where packets are buffered in nodes when a disconnection occurs between two nodes and there is no continuous path available between them, thus increasing packets delay [4]. The store-carry and then forward mechanism is used in these algorithms when a connection is established, thus increasing delivery ratio and considerably reducing the data loss rate [5]. This mechanism keeps messages in node buffers during disconnection time, and takes the advantage of node mobility feature to find appropriate nodes within different partitions in order to route messages toward their destinations.

New routing algorithms must benefit from history of node contacts and status of nodes in a network topology for routing decisions in order to achieve efficient decisions. Due to the lack of stable links in opportunistic networks, the memory overhead is high. In addition, since permanent links do not exist in these networks, the bandwidth of links becomes an important resource that must be fully utilized when a contact is made. Therefore, identifying potential intermediate carrier nodes based on the network knowledge is essential for messages. By efficiently utilizing the bandwidth for a given message, the opportunity to transmit other messages in the network can increase. These issues motivate us to design a new routing algorithm that resolves them.

The authors objective is to propose the Location and Direction Aware Opportunistic Routing (LDAOR) algorithm that considers position of vehicles to avoid flooding messages to all contacts and to limit replication rate. The aim is on reducing the overhead in addition to improving delivery ratio, delay, drop ratio, and the number of aborted messages. In addition, the angle between motion vector of adjacent nodes and the distance vector from neighbor node to destination node is considered to avoid sending a message to the vehicles moving in the opposite direction of the message destination.

The authors contribution is the proposal of LDAOR as a location-based and history-based knowledge OR technique that chooses the best forwarder node to carry a message to its destination based on the position and direction of the forwarder node with respect to the destination. In addition, it picks messages for transmission based on their assigned priorities to quickly forward messages to their destinations before overflowing of limited buffers used in nodes.

2. Related Works

The OR algorithms have several prominent functions as:

1. multi-copy,
2. single copy,
3. location-based,

4. history-based,
5. special node-based [6], [7].

Each one of the OR algorithms can take the advantages of one or several functions for making their routing decisions.

The first two functions are often used in flooding-based OR algorithms. Under the first function [8], [9], the copy of a message is given to all intermediate nodes connected to a carrier node. This causes a message to move towards its destination through many directions, thus increasing delivery rate. However, a message may be given to the vehicles not moving towards the destination node of the message making buffers full in nodes, losing a group of messages, and increasing network overhead. In addition, this function can seriously reduce the network efficiency under low network resources [7]. On the other hand, using the single copy function, the number of copies of a message is limited. Either a carrier node or the first node that communicates with carrier nodes attempts to directly deliver the copy of a message to its desired destination [10], [11]. Some techniques dynamically determine the number of required copies of a message according to network conditions [12]–[15]. The location-based algorithms take the advantage of physical location of vehicles for routing during establishing a connection [16]–[20]. These methods decide regardless of the status of nodes in the past, but their advantages in using updated information in nodes are consistent with network conditions. Based on the fourth function, the history of contacts is used for making routing decisions [9], [21], in which routing decisions are assessed according to general network conditions within the entire network and during all the time. However, right decisions may not be made due to old information. Finally, in OR based on special nodes [22]–[24], certain nodes are used to deliver messages to destination nodes.

Many algorithms have been introduced to reduce the flooding effects, e.g., [12]–[15], [25], [26], by forwarding a message to high-quality nodes that have better chance for delivering the message to its destination. The quality of a node can be defined by various metrics such as the frequency that a node meets other nodes, the frequency that a node meets the destination, the last contact time of a node with other nodes, and the last contact time of a node with the destination. For example, MaxProp [9] is a flooding-based protocol [12], [27] since a carrier node sends messages to all contacts without distinguishing between them. Besides, MaxProp is based on history [7] since it takes the advantage of history of contact nodes for prioritization of messages for transmission and for removing from node buffers. It prioritizes each message based on two criteria: hop-count, i.e., number of nodes a message has traveled since its generation, and delivery probability to destination. When sorted based on hop-count, the messages with the hop-count smaller than a given threshold have high priority for transmission. On the other hand, those messages with hop-counts exceeding the threshold are sorted based on delivery probability to their destinations. Then, the messages with small chance of delivery to their destinations have the highest priority to be deleted from the buffer when the buffer is becoming full.

Some OR protocols are location-based that can provide better performance results than [28]–[30]. Instead of links statuses, routing decisions are based on the positions of source node, destination node and adjacent nodes at contact time. Since location-based protocols do not require the overhead of saving the information of previous nodes, they can achieve the desired goal with minimal overhead. For example, in Packet-Oriented Routing (POR) [18], messages belonging to far destinations have higher priority for transmission compared to the messages belonging to close destinations. There is no limit on buffering messages in nodes under POR. The POR decides only based on the distance of an adjacent node to the destination node and does not consider the history of contacts at all. Using POR, a carrier node only selects the best forwarder node among adjacent nodes for all its messages. After prioritizing them, POR sends the messages in sequence to the new forwarder node.

The Prophet protocol [21] benefits from the history of contacts of nodes to destination node besides the multi-copy function. By this history, the delivery probability of a message to its destination through adjacent nodes can be computed by a carrier node. Then, the message will be given to those nodes that have visited the destination node more than the carrier node itself.

In this paper, the best forwarder nodes are selected using their statuses at the time of contact to prevent from flooding. In addition, based on the history of contacts and position of nodes in the network, priorities are assigned to messages for sending and removing them from buffers. This can improve performance parameters in the whole network.

3. The LDAOR Protocol

In the following, the LDAOR protocol shall be described after stating network model.

3.1. Network Model, Data Structures, and Performance Parameters

The following shows presented network model and its relevant assumptions:

- The network focuses on vehicle to vehicle communications in an urban area with many junctions in which roads are two-way. For example, Helsinki city includes these features.
- The proposed VANET includes different vehicles such as privately-owned vehicles (POV), buses and taxis with special mobility patterns. Among the vehicles, several cars are randomly determined as destination of messages and other are selected as source vehicles. Destination nodes are considered stationary

(fixed). This assumption is suitable for applications such as delivering a message to a base point as access point.

- Since the speed of nodes cannot exceed a limit in a city scenario, each node can find the location of a destination node using a suitable location server, e.g. the method presented in [32]. Location servers can provide lookup and publish algorithms to exchange information about geographic positions of nodes in a network. In order to select the best forwarder nodes, a carrier node needs to calculate the direction of any candidate node with respect to the destination node and its distance to the destination node.
- Each vehicle is equipped with Geographical Position Systems (GPS) to obtain its current position.
- Since there is no resource without limitation, each node has a limited buffer.
- Transmission rates in all vehicles are all the same.
- There is no faulty node or link in the network.
- There are two reasons for having low collision in opportunistic networks. First, the number of neighbor nodes that can correctly receive and send messages is low because of instability links. Second, since the positions of nodes are different, the nodes do not receive a request message from a carrier node at the same time.

The following list shows the data structures required for the LDAOR protocol:

- Node A uses five fields to keep its status as:
  - **Node ID** – identification code of node A;
  - **Delivery probability list** – the list of delivery probabilities relevant to delivering of messages from node A to each one of other nodes. For example, if the value of probability in carrier node B to node A is 0.25, node B has made contact to node A with probability 0.25 so far. There is only one delivery probability in node A to each one of other nodes in the network. When a contact is made between nodes A and B, delivery probabilities are updated in both nodes based on the method presented in MaxProp. In the same way, a carrier node may evaluate the probability of nodes to the destination node for finding a node on a path with a high contact probability. To calculate delivery probability, whenever node X makes contact to node Y, the value of probability in all nodes is increased by one, and then the probabilities of all nodes are re-normalized based on the rule provided in Section 3 of [9] so that sum of probabilities in all nodes becomes 1. Thus, delivery probability list values in a node indicate that this node moves in either a crowed path or a sparse path;
  - **List of MsgIDs already sent** – this list keeps the ID of transmitted messages to other nodes. For example, consider node A has sent messages M_1 and M_2 to node B; and M_3, M_4, and M_5 to node C. Then, node A keeps {B, (M_1, M_2)} and {C, (M_3, M_4, M_5)} in this field. In the next contact, node A can easily find out that node B has previously received message M_2, and it will avoid sending message M_2 to node B for the second time. This list is scanned at some time intervals and the old IDs are removed from the list;
  - **Current Location(x, y)** – at any time, current location of node A is obtained from the GPS system and saved in these fields;
  - **Previous location(x, y)** – previous location of node A;
  - **Average transmitted bytes per transfer opportunity** – when node A makes a contact with node B, and then sends its messages to node B within the contact period, total number of transmitted bytes in node A is updated by the size of transmitted messages.
- Each message M includes the following fields:
  - **MsgID** – identification code of message M given by the source node. This ID is a combination of node ID and a sequence number generated by the node;
  - **Source** – the ID of the source node that has generated message M;
  - **Destination** – the ID of destination node of message M;
  - **Hop list** – when message M passes through different intermediate nodes, the IDs of the intermediate nodes are recorded in this list;
  - **TTL** – time to live for message M. When TTL of the message expires, the message should be deleted.
  - **Hop-count** – This field is set to zero when message M is generated. When this message arrives at an intermediate node, the hop-count field is incremented by 1;
  - **Message text** – the text of message M.

In a Hello message, there is a field called AckedMsgs. To avoid propagating a message already reached its destination, the destination node adds the ID of the delivered message to the AckedMsgs field of a Hello message and broadcasts it in the network. Then, each node receiving this Hello message removes the acknowledged messages from its buffer. The following shows the performance parameters defined based on the ONE simulator [35]:

- **Aborted messages** – the number of aborted transmissions between nodes divided by total number of generated messages. A message is aborted when a receiver node cannot receive the message from a transmitting node because of small contact duration.
- **Loss in buffers** – number of dropped messages (including replicated messages) in buffers due to buffer overflow. This loss occurs when a receiving node does not have enough room in its buffer.
- **Delivery ratio** – message delivery probability defined by

$$Delivery\ ratio = \frac{Number\ of\ delivered\ messages}{Number\ of\ generated\ messages}.$$

- **Overhead ratio** – assessment of bandwidth efficiency defined by

$$Overhead = \frac{Number\ of\ relayed\ messages\ including\ replicated\ message}{Number\ of\ delivered\ messages} - \frac{Number\ of\ delivered\ messages}{Number\ of\ delivered\ messages}.$$

  Since in all OR algorithms request messages and reply messages should be communicated between intermediate nodes, their overhead has not been considered in the overhead ratio. According to the above formula, only those data messages that cannot be delivered to their destinations are accounted for overhead ratio.

- End-to-end delay – average delay from generation time of a message until successfully delivering to its destination.

3.2. The LDAOR Protocol

The pseudo code of LDAOR is shown in Algorithm 1 and Algorithm 2. It includes two phases: selecting the best neighbor nodes in order to store-carry and then forward a message toward its destination (see Subsection 3.2.1), and prioritization of messages according to MaxProp and then sorting the messages based on the minimum distance between neighbor nodes and destination nodes (see Subsection 3.2.2). To reduce overhead, LDAOR limits the rate of message replication with respect to physical location and direction of neighbor nodes with the destination node. The LDAOR utilizes both node history information and node information at the time of contact. The general parameters used in LDAOR are depicted in Table 1.

Algorithm 1: LDAOR, phase 1 – neighbor selection on contact event

```
Step 1: Exchange the status of connection to each other
Step 2: Delete the acknowledged messages from the buffer
Step 3: Direct delivery: if the neighbor node is the destination
of any message in the buffer, then deliver it first
Step 4: CM = {}
    for each message M_k in carrier node c_i do
        n_k = number of neighbors that have not received M_k
        if (n_k = 0) then
          M_k must still remain in buffer
        if (n_k = 1) then
          g_n = best forwarder node based on the angle-based
          method
        else if (n_k > 1) then
          g_n = best forwarder node based on the distance-
          based method
        end if
          if (n_k ≠ 0) then add (g_n, M_k) to set CM
    end for
    return CM
```

Algorithm 2: LDAOR, phase 2 – determine priorities of messages for transmission

```
Step 5: Determine threshold H   // similar to MaxProp
Step 6: Split CM into two sections – sorting messages with
hop-count lower than threshold H and messages with hop-count
greater than threshold H
Step 7: for k = 2 to size (CM) do   // take all messages in CM
        Take messages M_k and M_{k-1} from set CM
          if (h_k < H and h_{k-1} ≥ H) then
          send M_k
          else if (h_{k-1} and h_k ≥ H) then
          1. if (h_k < H and h_{k-1} < H) then
               if (DI_k < DI_{k-1}) then Send M_k
               else                 Send M_{k-1}
               end if
          2. if (h_k ≥ H and h_{k-1} ≥ H) then
               if (dp_k > dp_{k-1}) then Send M_k
               else if (dp_k < dp_{k-1}) then Send M_{k-1}
                 else   // the same dp for forwarders of M_k
                          and M_{k-1}
                   if (DI_k < DI_{k-1}) then Send M_k
                   else Send M_{k-1}
                 end if
          end if
      end for
```

3.2.1. Neighbor Node Selection (Phase 1)

Whenever a carrier node wants to find a forwarder node, it should select the best neighbor node as a forwarder node in phase 1.

Step 1 of phase 1. The operation of the first phase of the algorithm is as follows. Since LDAOR decides for each message individually, it is necessary to obtain required information from its adjacent nodes in order to select the

Table 1
Notations used in LDAOR

Notation	Description
angle	Angle between two location coordinates for a neighbor node
b	Buffer size
c_i	Carrier node i
c_n	Candidate neighbor node
CM	Array of selected forwarder nodes with messages for transmission
DI_k	Distance array between neighbor nodes k and destination node
di_c	Distance value between candidate node c and destination node
d_k	The destination of the k-th message
dp	Delivery probability for a neighbor node
g_n	Good neighbor
H	Threshold on hop-count of a message
h_k	Hop-count of the k-th message
(lc_x, lc_y)	Location coordinate for neighbor node at current time
(lp_x, lp_y)	Location coordinate for candidate neighbor node at previous time
M_k	The k-th message in a buffer
$\overrightarrow{ND}$	Distance vector from neighbor node to destination M_k
n_k	Number of neighbors that have not received M_k
p	Portion of buffer
speed	Speed of neighbor node c_n
S_T	Average transmitted bytes
$\vec{V}_N$	Neighbor node velocity vector
(v_x, v_y)	Velocity coordinate of a neighbor node
θ	A direction angle of a neighbor node

best forwarder node among them when a contact happens. This information includes: the acknowledged messages in order to avoid decision making once again and resending them again, and the location and direction of neighbor nodes in order to check their statuses with respect to the destination of messages, where LDAOR obtains this information using location server. These operations are carried out based on the information available in nodes (see Subsection 3.1).

Step 2 of phase 1. After receiving the acknowledge message for a transmitted message, a carrier node removes the message from its buffer. Notice that relative mobility between two vehicles may be high. Then, there will be a delay in receiving acknowledge of messages. In this case, the carrier node removes the acknowledged messages from its buffer using the following mechanism. If the carrier node has not received the acknowledge for a transmitted message, it assumes that the message has not been delivered to the relevant destination node yet. Therefore, it tries to find a forwarder node for carrying the message by sending a request message (in order to resubmit the message) to neighbor nodes. It is likely that some of the neighbor nodes have already received the acknowledge of the message. Hence, they avoid receiving the duplicate message, and notify the carrier node (with a reply message) that the message has already been delivered to its destination. In the worst case, the message may be delivered to a node that has not received its acknowledge. Nevertheless, it is likely that its neighbor nodes have already received the acknowledge.

Step 3 of phase 1. If there is a neighbor node which is the destination of a message in the buffer, LDAOR directly delivers the message to the neighbor node. By this way, the number of messages from the buffer will reduce.

Step 4 of phase 1. If there is no destination node for a message among the neighbor nodes, LDAOR enters the decision making step for selecting the best neighbor nodes for carrying the remaining messages inside the buffer of the carrier node (see Fig. 1).

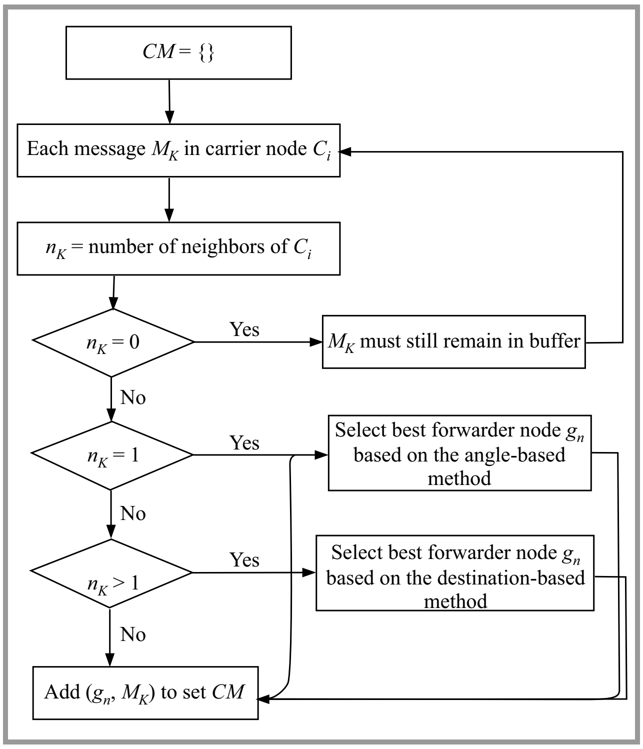

Fig. 1. Step 4 in phase 1.

The neighbor node selection mechanism in LDAOR operates as follows, in which only one forwarder node can be selected for each message. For a given message, when there is only one contact that has not previously received the message, the carrier node uses the angle-based method. On the other hand, when the carrier node has several contacts that neither of them has previously received the message, the distance-based method is utilized for forwarder node selection. Note that $n_k = 1$ in a region indicates that the number of candidate nodes is low and there is the probability of partitioning. This is because in this region, only one node has been candidate to receive the message. Thus, sending the message to this node needs more precision. Hence, if the message is sent to a node that moves away from the destination node (in the worst case, it moves in the opposite direction to the destination node), the chance of sending the message to the best other node is low, and therefore, this transmission may be vain in the network. However, $n_k > 1$ indicates that congestion of nodes is relatively high in the region, and the closer node to the destination node for sending the message is better. In this case, even if this node moves away from the destination node, there are some chances for delivering the message to the other best node. At the end, set *CM* is provided, where each item in this set includes two entries as the selected neighbor node and the relevant message.

To illustrate the importance of neighbor node selection, consider the scenario displayed in Fig. 2a. Suppose carrier node A has messages M_1, M_2 and M_3 in its buffer, respectively, with destinations D_1, D_2 and D_3. Node A has made contact with three nodes B, C and F. Node A has received the status of the nodes in response to Hello messages for carrying message M_1. Then, based on the status of nodes, node A is noticed that node B has previously received message M_1. Thus, either node C or node F should be selected to carry message M_1. Using the distance-based method, node F with minimum distance to destination D_1 is chosen as the forwarder node for message M_1. Based on the statuses of adjacent nodes, node A is also noticed that only node C has not previously received message M_2. Hence, node A uses an angle-based method to check whether node C moves toward destination D_2 or not. After calculating the angle between motion vector of neighbor node C and distance vector of neighbor node C to D_2, node A realizes that node C moves completely in opposite direction to destination D_2. Thus, M_2 has no chance to be delivered to D_2 by node C. Therefore, node A must still keep M_2 in its buffer until the setup of an appropriate contact. This avoids useless saving of a message in buffer of node C. Similarly, the carrier node chooses the best forwarder node for other messages in its buffer. After preparing the set of ready contacts for receiving messages (as depicted in Fig. 2b), the phase 2 is started. The carrier node sends the messages based on their priorities to the appropriate nodes selected in the first phase (see Subsection 3.2.2).

When a carrier node wants to select the best forwarder node, it uses the location service for receiving the location of the destination node. The carrier node decides to select either the distance based method or the angle-based method according to the number of candidate nodes. In the distance based method, the carrier node obtains the position of the destination node by location service in order to compute the distance of the candidate node to the destination node. In the angle-based method, the carrier node obtains the

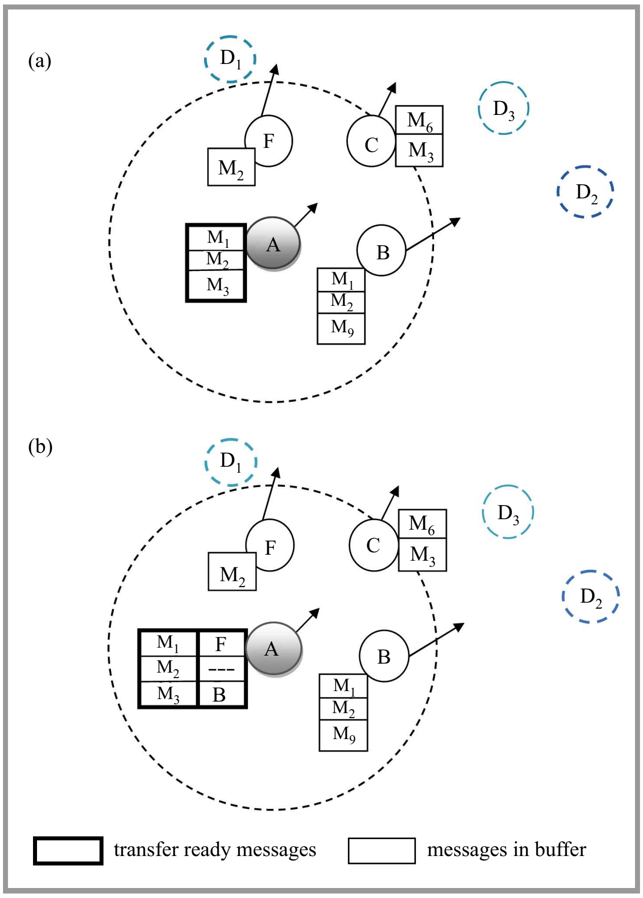

Fig. 2. Neighbor selection scenarios: (a) neighbor node selection scenario in LDAOR and (b) contacts with messages ready to transmit.

position of the destination node by location service in order to compute the angle between the distance vector to the destination node and the motion vector of the candidate node.

In angle-based method an exactly prediction of direction of vehicles is more complex in city scenarios due to high number of branches. Therefore, LDAOR determines the direction of a neighbor node by approximating the angle between the velocity vector of the neighbor node and distance vector from the neighbor node to the destination node (see Algorithm 3). If there exists any calculation error for the angle, it surely depends on the received information from the nodes that may depend on the error originated from GPS. Therefore, if there is a GPS with low error, LDAOR can calculate the angle according to the Eq. (1) very well.

In this case, if the estimated angle is acute angle (i.e. below 90°), then the neighbor vehicle is likely moving toward the destination node. Actually, in LDAOR, vehicle direction with respect to the destination node is important, not its direction in each junction.

Let vehicle V be the only candidate node for receiving the message. Then, the carrier node uses the angle-based method for approximating the direction of vehicle V with respect to the message destination. In this case, the mes-

Algorithm 3: Angle-based method to evaluate a neighbor node c_n for a given message M

$angle = \tan^{-1}\left(\frac{lc_y - lp_x}{lc_x - lp_x}\right)$
$v_x = speed \times \cos(angle)$
$v_y = speed \times \sin(angle)$
$|\vec{V}_N| = \sqrt{v_x^2 + v_y^2}$
Compute based on destination of message M
$\theta = \cos^{-1}\left(\frac{\overrightarrow{ND} \times \vec{V}_N}{|\overrightarrow{ND}| \times |\vec{V}_N|}\right)$
if $(\theta < 90^\circ)$ **then**
  c_n is chosen as forwarder node for the message
else
  The message should still be kept in buffer
end if

sage is sent to a neighbor node V only if it moves toward the message destination. Determining whether node V is moving toward the message destination or not can be obtained by calculating θ:

$$\theta = \cos^{-1}\left(\frac{\overrightarrow{ND} \times \vec{V}_N}{|\overrightarrow{ND}| \times |\vec{V}_N|}\right), \quad (1)$$

where $\overrightarrow{ND}$ is the distance vector from node V to the destination node of the message, and $\vec{V}_N$ is the velocity vector of node V. If $\theta < 90^\circ$, node V is likely moving toward the message destination, i.e., the message can be delivered to it. Otherwise, when $\theta \geq 90^\circ$, node V moves away from the destination node, and therefore, the message should be held in the carrier node buffer until finding a better forwarder node. This mechanism avoids delivering messages to the nodes that do not move toward the destination. Therefore, the network traffic decreases and buffers are not filled with those messages that do not have any chance to be delivered to their destinations.

Algorithm 4: Neighbor selection by distance-based method for message M_k with destination d_k

D = infinity
for each neighbor node c_n **do**
  $di_c = \text{distance}(c_n, d_k)$
  if $(di_c < D)$ then
    $D = di_c$
    $g = c_n$
  end if
end for
return g

In distance-based method the multiple nodes are candidate for receiving the message. In this case, the carrier node uses the distance-based method (see Algorithm 4) for selecting the best forwarder node. Hence, each candidate neighbor node notifies its physical location inside a reply message to the carrier node. Then, the carrier node computes the distance of each candidate vehicle from its current position to the destination node and selects the node with the smallest distance to the destination node as the forwarder node. Then, the carrier node delivers the message to the selected forwarder node.
In short, among multiple candidate neighbor nodes, a node is selected with the minimum distance to the destination node of a given message. This method can relatively remove redundancy created by the flooding methods. In addition, delivering the message to the nodes that are closer to the message destination can reduce delay.

3.2.2. Management of Buffer in a Carrier Node (Phase 2)

Since it is assumed that nodes have limited buffers, their overflows may happen and some important messages may be lost. Hence, a mechanism should be provided for buffer management so that the messages that are more likely to reach their destinations are processed faster. On the other hand, those messages with minimum chance of delivery to their destinations should be removed from buffers when overflowing, i.e. the messages with minimum delivery probabilities with hop counts exceeding a threshold. These deleted messages are counted as lost messages. Therefore, messages should be prioritized in buffer of each node. By this, more space can be provided for future coming messages. In LDAOR, transmission opportunity is the same for all messages at the beginning. However, when message M_1 has been transmitted for a number of times (i.e. a message with high hop-count), message M_1 should have less priority in re-transmission compared with newly arrived message M_2. This is because message M_1 has already used network bandwidth and has not been successfully delivered yet. Hence, it is fair to service message M_2. Still message M_1 has transmission opportunity in future. The LDAOR tries to provide fairness for message transmission opportunity. When two messages have the same hop count, decision is made based on the status of the transmitter node (as stated in the following). Therefore, messages will not encounter bandwidth starvation under LDAOR.

Step 5 of phase 2. Consider a carrier node has made contact with a given node. Under LDAOR, the carrier node can send its traffic to the given node as long as the contact is setup. Whenever the contact is shut down, the carrier node computes the volume of transmitted traffic in that contact in bytes. Note that, within the contact period, the carrier node may transmit a number of messages or only a part of a message. Then, the carrier node computes average transmitted traffic S_T among contacts as following. Consider a sliding window of 10 last contacts set up by the carrier node. Let S_i be the volume of whole traffic transmitted in bytes within the i-th contact in the sliding window, computed at the end of the contact i. Let r (where $r \leq 10$) denote the number of contacts made within the sliding window. Then, S_T is computed by

$$S_T = \frac{1}{r}\sum_{i=1}^{r} S_i, \quad \text{if } S_i \neq 0.$$

Prioritization of messages in LDAOR is performed by the following rules. For prioritization of messages, we need to define parameter H as a threshold on hop-count of messages in a given carrier node, which is the same for all messages inside the carrier node buffer. Note that each node has its own H at any time. This threshold is computed based on the messages available in the carrier node buffer. Similar to MaxProp, the LDAOR uses average transmitted bytes S_T and buffer size b to adjust threshold H. For this purpose, the carrier node calculates parameter p using Eq. (2) (in bytes) [9]:

$$p = \begin{cases} S_T & S_T < \frac{b}{2} \\ \min(S_T, b - S_T) & \frac{b}{2} \le S_T < b \\ 0 & b < S_T \end{cases} . \quad (2)$$

A carrier node computes parameter p in Eq. (2) under two situations since average number of transmitted bytes S_T may have changed:

- When the carrier node wants to send a message, it must calculate p according to average transmitted bytes S_T. Then, it sorts the messages in its buffer based on their hop counts. Finally, the node calculates threshold value H;

- When the carrier node wants to receive a message while its buffer is full, it needs to re-calculate p. Then, considering the threshold value, it removes a low-priority message from its full buffer in order to receive the new message.

Then, z items in set CM (obtained in phase 1) are sorted (in ascending order) based on the hop-count of the messages. Starting from the beginning of the sorted list CM, denote the size of z messages by $N_1, N_2, \ldots, N_z$. Now consider the j-th message satisfies the condition $\sum_{i=1}^{j} N_i > p$, where $\sum_{i=1}^{j-1} N_i \le p$. Then, threshold H is set to the hop-count of the j-th message plus 1.

After computing threshold H, the messages in CM are split into two sections (similar to MaxProp): messages with hop-count $< H$ and messages with hop-count $\ge H$. The LDAOR mechanism determines the priorities for transmission of messages and deleting messages from the carrier node buffer as displayed in Fig. 3. Notice at the left part of Fig. 3, messages are first sorted based on hop-count, and then based on distance of neighbor nodes to destinations. In other words, if hop-counts of few messages are the same, they are sorted based on the distance of their neighbor nodes to their destination nodes. On the other hand, the right part in Fig. 3 is sorted based on delivery probability. When delivery probability is the same for a few messages, they are sorted based on distance of neighbor nodes to destination nodes, as displayed in Fig. 3.

As stated in Subsection 3.1, hop-count of a message shows the number of nodes the message has visited. By sorting messages based on their hop-counts in a carrier node, authors give high priority of transmission to those messages that have been generated newly and have visited small

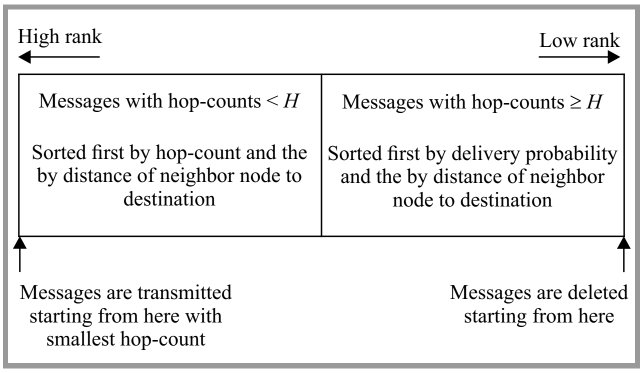

Fig. 3. The LDAOR priority mechanism for splitting messages in *CM*.

number of nodes. Notice a message with high hop-count shows that the network has attempted for a number of times to deliver the message to its destination by passing through a high number of nodes. However, the message has found less chance of delivery and less importance.

Delivery probability for a forwarder node shows the number of frequencies that the forwarder node has made contacts with other nodes. When delivery probability is high for a forwarder node, it shows that the delivering of messages through that forwarder node is high. When a contact is made between two nodes, delivery probabilities are updated in both nodes, based on the method presented in [9].

Step 7 of phase 2. In this step, di_c shows the distance between the selected forwarder node and the destination node of message M_k, the notation h_k denotes the hop-count of message M_k, and dp_k refers to the delivery probability of the forwarder node assigned to message M_k. In each loop, Step 7 evaluates two consecutive messages M_k and M_{k-1} in CM. If hop-counts of both messages are smaller than threshold H, their priorities are determined based on the distance-based method, i.e. high priority for transmission is given to the message that its relevant forwarder node has smaller distance to its destination. The advantage of this mechanism is that when a forwarder node has the smallest distance to the destination, its relevant message is transmitted at first. Then, the next messages are transmitted in sequence according to distances of their associated forwarders to their destinations. This mechanism causes a message to be located closer to its destination before its corresponding forwarder node goes in a situation that is out of the communication range of the destination. For example, consider there are three messages M_1, M_2, and M_3 in CM, where forwarder nodes c_{n1}, c_{n2}, and c_{n3} have been assigned to carry them, respectively. Consider hop-counts of messages M_2 and M_3 are smaller than H, and hop-count of message M_1 is larger than H. Assume c_{n3} to be closer to destination of message M_3 than c_{n2} to destination of message M_2. Therefore, M_3 is transmitted sooner than M_2. After sending M_2 and M_3, the carrier node sends M_1 due to its hop-count greater than H. This mechanism can increase delivery rate and can reduce delay in delivering messages to their destinations.

Table 2
Example for message transmission

Hop count	8	2	12	18	5	3	8	10	11	16	9	3
msgID	2	9	11	8	4	13	5	7	1	19	16	23
Message size	32099	55999	16558	432233	486776	52641	375477	189610	300743	336101	86929	254886

Table 3
Sorted messages

Hop count	2	3	3	5	8	8	9	10	11	12	16	18
msgID	9	13	23	4	2	5	16	7	1	11	19	8
Message size	55999	52641	254886	486776	32099	375477	86929	189610	300743	16558	**336101**	432233

On the other hand, for those messages with hop-counts exceeding threshold H, the LDAOR assesses the delivery probabilities of forwarder nodes to destination nodes. Then, the message with the highest delivery probability is sent to its forwarder node at first. Then, other messages with smaller delivery probabilities are sent in sequence. If delivery probability of forwarder nodes of both messages is the same, high priority for transmission is given to the message that its relevant forwarder node has smaller distance to its destination than the other forwarder node.

Let us consider as an example that there are 12 messages available in the buffer of a carrier node (see Table 2), and the carrier node has decide to send a message with the highest priority. Let average transmitted bytes be $S_T = 3,047,877$, and buffer size $b = 5,000,000$ bytes. Since $b > S_T > \frac{b}{2}$, we have $p = \min(S_T, b - S_T) = 1,952,123$ bytes. Then, the messages are sorted based on their hop-counts (see Table 3). Since the summation of messages sizes until the message with msgID=19 is greater than p, the hop-count of this message is chosen as our threshold with $H = 16 + 1 = 17$. Now all the messages in the buffer are split into two groups as depicted in Table 3, messages with hop-count smaller than H at the left side of buffer, and the messages with hop-count greater or equal than H at the right side of buffer.

3.2.3. Complexity Analysis of LDAOR and other OR Algorithms

The amount of transmitted information between nodes in LDAOR is almost similar to MaxProp because it uses the same mechanism for determining threshold as MaxProp, but LDAOR considers location of nodes as well. Prophet considers only the history of nodes contacts with the destination node for determining both the best forwarder node and priority of sending messages. The POR evaluates only the location of each node with respect to the destination node for determining both the best forwarder node and priority of messages.

Based on the distance-based and angle-based methods, a carrier node at first evaluates only the value of current location(x, y) and previous location(x, y). If a candidate node is chosen as the best forwarder node, the carrier node uses average transmitted bytes per transfer opportunity and delivery probability list for prioritizing messages. If a carrier node selects a candidate node, it enters the message prioritizing step and evaluates these two parameters. These values are based on the history of contacts and history of sending messages by this carrier node. It does not depend on a special time, and therefore, it cannot be outdated. The List of MsgIDs already sent is only kept in a node and it is not transmitted between nodes.

The computational complexity of Epidemic is O(1) [8] due to the lack of using any knowledge from the network. There is no forwarder node selection in MaxProp and it only prioritizes messages for transmission or removing from buffer. Hence, the complexity of MaxProp is $O(m \times \log_2(m))$, where m is the number of messages available in a node buffer. The MaxProp performs sorting twice for determining priority of messages. First for all nodes based on hop-count, and then based on delivery probability for the right part of a buffer.

Due to the process of selecting neighbor nodes in LDAOR, its complexity is $O(d)$ at phase 1, where d is the number of neighbor nodes for a message. LDAOR also adjusts prioritization for sending and removing a message with complexity $O(m \times \log_2(m))$ in phase 2. Notice LDAOR performs sorting twice; first, for all nodes based on hop-count, and then based on delivery probability for the right part of a buffer. Since for each message, only one neighbor node is selected, the complexity of LDAOR becomes $O(d \times m \times \log_2(m))$. Number of adjacent nodes d is small and this means that the nodes that can send or receive messages correctly is small because of instability of links [33], [34]. This is the most important distinction in opportunistic networks compared to other networks. In other words, the number of contacts is low, i.e. the number of neighbor nodes, and not the number of nodes in total [33], [34]. This is why there is a store-carry and forward mechanism in opportunistic networks.

Similar to LDAOR, the POR checks each one of its adjacent nodes for selecting the best forwarder node and also determines priorities for messages. Thus, the complexity of POR is the same as LDAOR. The Prophet counts the number of times adjacent nodes have visited a given destination node. Then, those nodes that have met the destination node of a given message more than the carrier node

Table 4
Complexity of opportunistic routing algorithms

LDAOR	MaxProp	POR	Epidemic	Prophet
$O(m \times log_2(m) \times d)$	$O(m \times log_2(m))$	$O(m \times log_2(m) \times d)$	$O(1)$	$O(m \times log_2(m) \times d)$

itself are chosen as forwarder nodes. Next, the messages are sorted in a descending order based on the number of visits whose forwarders have met their destinations. Finally, the messages are transmitted from the sorted list. For example, consider there are three messages M_1, M_2, M_3 in the buffer of a carrier node, and the carrier node has found that neighbor nodes c_{n1}, c_{n2}, and c_{n3} have met the destinations of M_1, M_2, and M_3, respectively, three, two, and five times more than the carrier node itself. Then, the carrier node first sends M_3, followed by M_1 and M_2. Hence, the complexity of Prophet is also $O(d \times m \times log_2(m))$. The complexities of the algorithms are shown in Table 4.

Notice that updating transmitted bytes and computing the threshold value is performed by simple arithmetic operations such as comparison and subtraction with complexity $O(1)$. In addition, the complexity of sorting messages in a buffer is $O(m \times \log_2(m))$, where m is number of messages in the buffer. For example, average contact ratio per hour (obtained from 30 times simulation replications) is 396.1364 in LDAOR and 395.7455 in MaxProp. Even in the worst case, if all contacts send messages or delete messages, it is not so big complexity for today's advanced processors.

Although the complexity of LDAOR seems to be higher than the complexity of MaxProp and Epidemic, the number of adjacent nodes is usually small in opportunistic networks and $O(d \times m \times \log_2(m))$ can be approximately considered as $O(m \times \log_2(m))$, especially at low density traffic.

4. Performance Evaluation

The performance of LDAOR is compared with Prophet [21], POR [18], Epidemic, and MaxProp [9] in urban environments. The conducted performance evaluation is based on the network model stated in Subsection 3.1. Simulations are performed using the Opportunistic Network Environment (ONE) simulator [35], an open source Java-based simulator designed for evaluation of opportunistic networks and DTN routing algorithms.

Fig. 4. The Helsinki city scenario in [33].

To approach a real environment, three different types of vehicles (private vehicle, bus, and taxi) with specific mobility patterns are considered in the Helsinki map (see Fig. 4). This map matches to considered urban features stated in Subsection 3.1. Since Helsinki is one of the cities that have good features including more branches (junctions), it is widely used as benchmarking city in many articles, e.g. [31], [36]–[39]. Private vehicles move based on the Map-Based model developed in the ONE simulator. In this model, each private vehicle randomly chooses a path based on the city map to reach its destinations. Buses follow predefined routes based on the Bus Movement model so that when a bus reaches the end of its path, it moves back to the beginning of the path. Similar to buses, taxis move on predefined routes. Unlike a bus, a taxi can choose the shortest path between the source and destination. Recall several vehicles are randomly selected as destination of messages and other vehicles are selected as source vehicles. Note that destination nodes can be considered as

Table 5
Parameter settings in ONE simulator

Network simulator	ONE
Simulation area	4500×3400 m
Simulation duration	12 h
Message size	Uniform (800 B, 500 KB)
Buffer size	5 MB
Data rate	2 Mb/s
Message TTL	300 min
Transmit range	200 m
Average speed	Uniform (10, 50) km/h
Number of nodes (taxi, bus, POV)	Dynamic (see Table 5)
Mobility model	Taxi: MapRoute Movement
	Bus: Bus Movement
	POV: MapBased Movement

Table 6
Number of different vehicles

Number of nodes	20	40	60	80	100
Bus	1	2	3	4	5
Taxi	2	4	6	8	10
POV	17	24	51	68	85

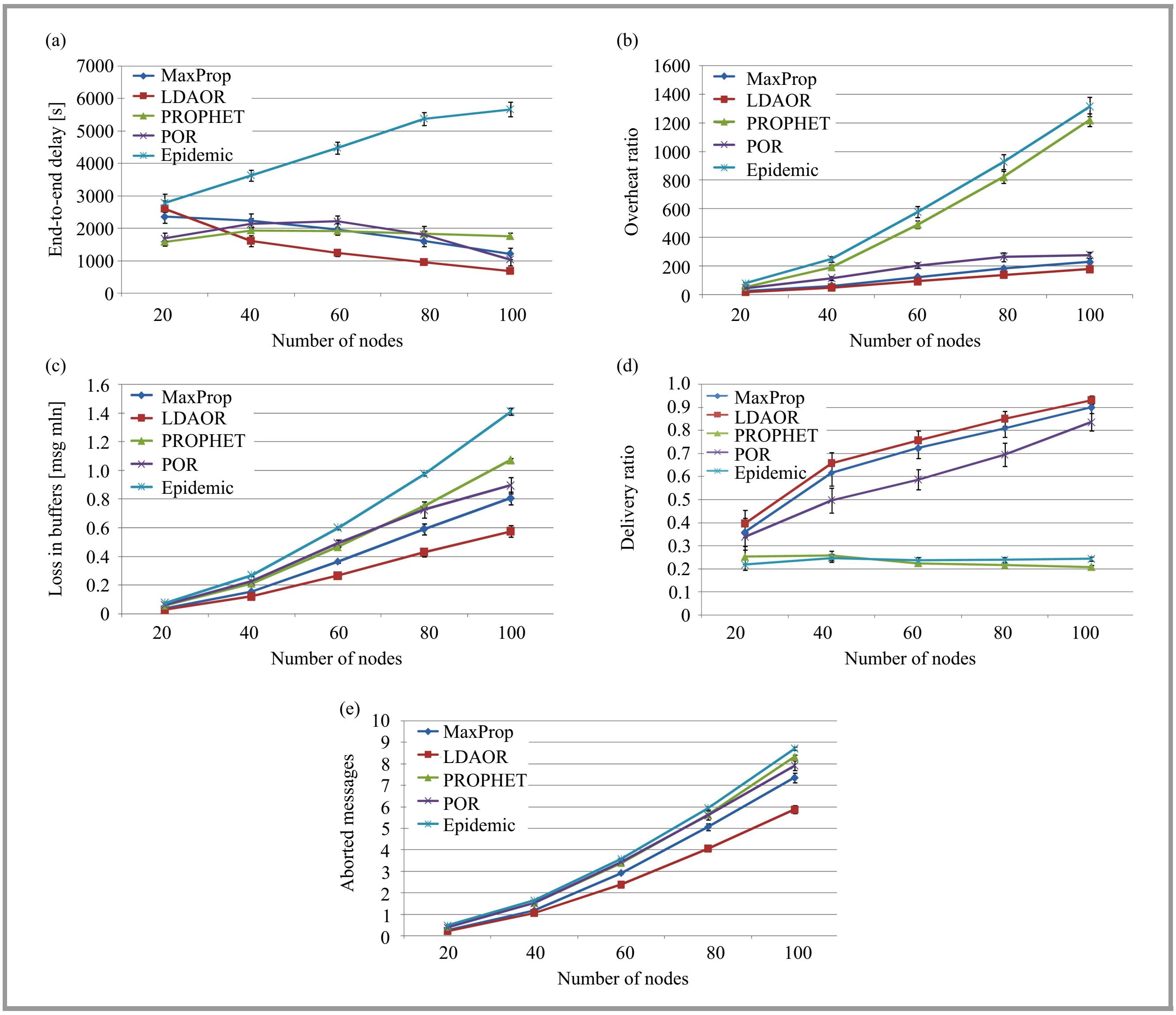

Fig. 5. Traffic load of 1 packet/ Uniform(5, 15) sec under different densities: (a) end-to-end delay, (b) overhead ratio, (c) loss in buffers, (d) delivery ratio, (e) aborted messages.

intermediate nodes for sending or receiving a message to other destination nodes, and therefore, the number of intermediate nodes is not less in the network.

Simulation parameters are shown in Table 5 and the numbers of different nodes are depicted in Table 6. In each diagram, simulation results are plotted with 95% confidence interval, where for each point 30 simulation replications have been done. Performance diagrams shown in the following are all measured within simulation period of 12 hours.

Note that the original POR paper has not considered any limitation for buffer of nodes. But, for simulating the algorithms under the same situation in presented simulations, authors consider buffer limitation for POR.

In the following evaluations, traffic load is expressed as the

$$\frac{\text{Arrival of } x \text{ packets}}{\text{Uniform } (y, z)}.$$

For example, in traffic load of one packet per Uniform (5, 15), inter-arrivals are computed from Uniform (5, 15) and in each inter-arrival one packet arrives at a node. Similarly in traffic load of 5 packets per Uniform (1, 2), inter-arrivals follow the distribution of Uniform (1, 2) sec and in each inter-arrival five packets arrive as a burst in a node.

The diagrams in Fig. 4 show the performance results under traffic load of one packet/Uniform (5, 15) sec. As one can see in Fig. 5a, when the density of the network is very small, LDAOR has more end-to-end delay than Prophet, POR, and MaxProp. This is because the number of nodes that meet the criteria defined by LDAOR is small. However, by increasing the density of vehicles in the network, end-to-end delay of LDAOR declines so that LDAOR achieves the best end-to-end delay when number of nodes becomes greater than 40. Figure 4b shows that not only LDAOR has the smallest overhead, but also its overhead is relatively

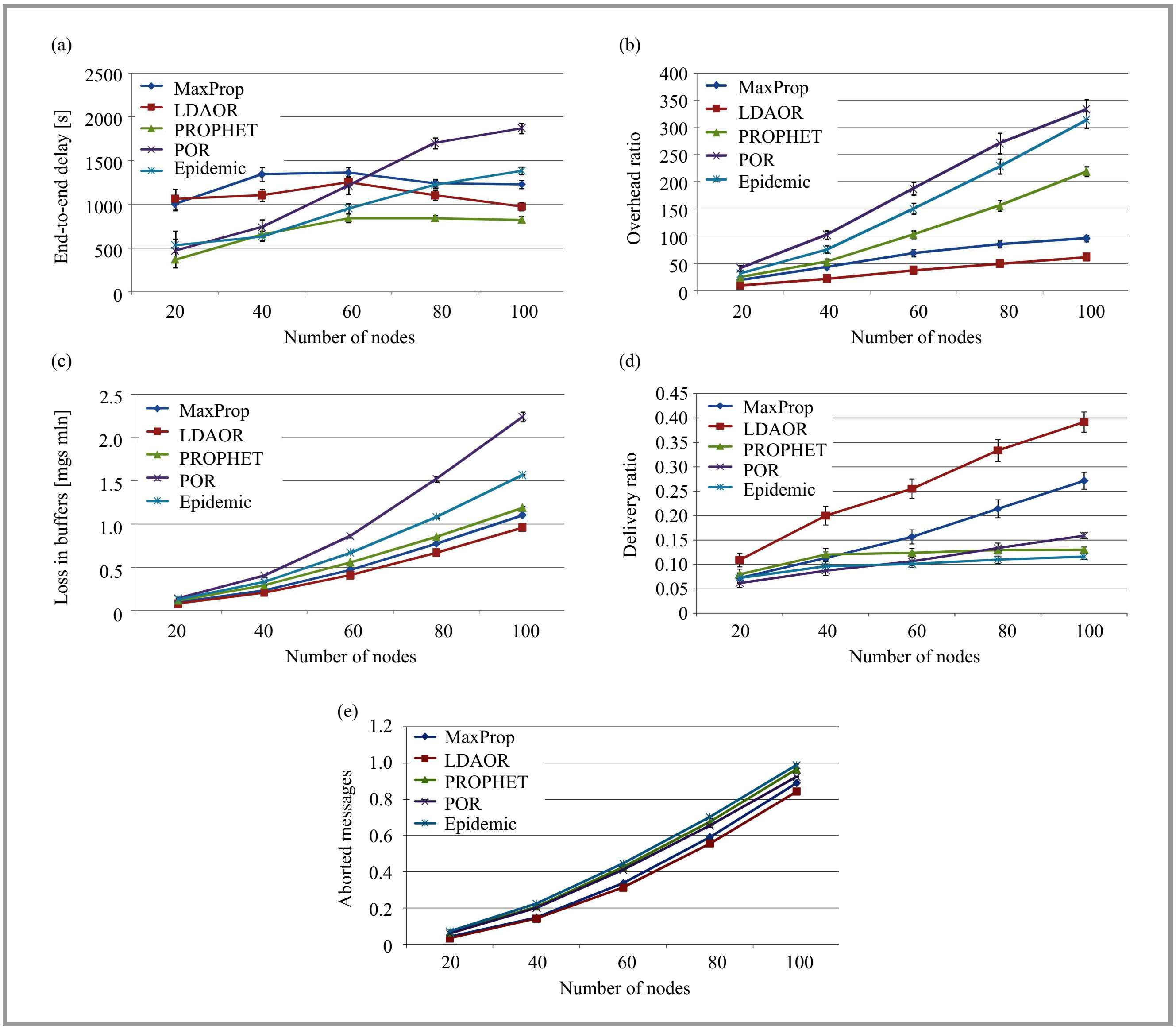

Fig. 6. Traffic load of one packet/Uniform(1, 2) sec under different densities: (a) end-to-end delay, (b) overhead ratio, (c) loss in buffers, (d) delivery ratio, (e) aborted messages.

flat. However, when number of nodes is very small, there is almost no difference among the overhead of LDAOR, POR, and MaxProp algorithms. This improvement on overhead is due to the criteria defined by LDAOR to avoid flooding.

Results in Fig. 5c illustrate the traffic loss from buffers when they overflow. The loss in LDAOR is less than other routing algorithms under different network densities. This is because of the fact that LDAOR mostly sends a message to those nodes that have much chance of delivering the message to its destination. Hence, the buffers are not completely filled in vain. Therefore, there will be enough space in buffers for saving those messages that have chance of delivery to their destinations.

Notice that number of relayed messages (and as a result the number of aborted messages) is more than the number of generated messages due to message replication. In addition, the number of dropped messages (loss in buffers) includes replicated messages. Hence, the values displayed in Fig. 5a,c,e are high.

At the first sight it seems that flooding-based OR algorithms should provide the highest delivery rate. However, as Fig. 5d depicts, LDAOR not only has increased the delivery rate but also has avoided flooding of messages. The delivery rate is rising when increasing network density. The main reasons for increasing the delivery rate under LDAOR compared to other algorithms are:

- a message in a buffer is only sent to those neighbor nodes located in better positions with respect to the message destination. Therefore, by preventing from additional transmissions and receiving of messages, bandwidth can be efficiently utilized and the opportunity for transmitting messages increases;
- priority for transmission of messages is provided based on the physical locations of new forwarder

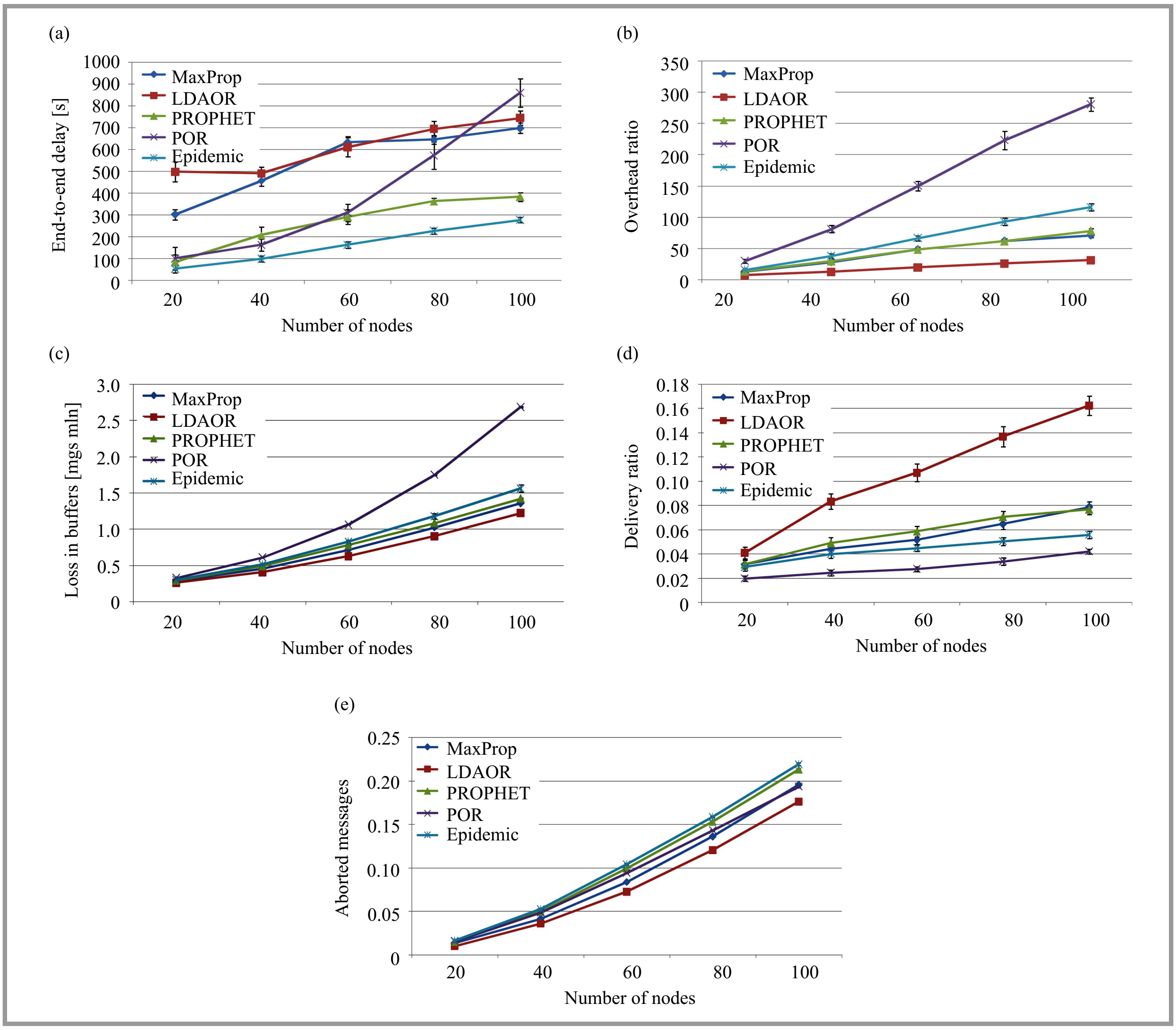

Fig. 7. Traffic load of 5 packets/Uniform(1, 2) sec under different densities: (a) end-to-end delay, (b) overhead ratio, (c) loss in buffers, (d) delivery ratio, (e) aborted messages.

nodes in addition to the history of contacts. By this, neighbor nodes (with high delivery probability) closer to the destination of a message can be selected as forwarder nodes of that message.

In the following, performance evaluation is performed at relatively higher traffic loads compared with Fig. 5. The diagrams in Fig. 6 and Fig. 7 show performance results under traffic load of one packet/Uniform(1, 2) sec and 5 packets/Uniform(1, 2) sec. By increasing traffic load, significant differences can be achieved compared to Fig. 5. For all routing algorithms, Fig. 6a and Fig. 7a depict significant reduction in end-to-end delay compared to Fig. 5a. Notice that by increasing traffic load, opportunity for transmission of all messages saved in a buffer decreases, thus reducing message delivery rate. Since message delivery ratio decreases, only those messages that are easy to be delivered quickly arrive at their destinations, thus reducing delay. Recall end-to-end delay is only averaged over successfully delivered messages. As it can be observed, end-to-end delay under LDAOR is more than some protocols at high traffic loads because determining a suitable node for each message leads to more waiting time in buffers.

Results in Fig. 6b and Fig. 7b show that POR has more overhead than other protocols. This is because POR only chooses one forwarder node for all messages in a carrier node buffer while their destinations could be different. Although many messages are sent under POR, there may be no chance for successfully delivering some of them by the selected forwarder node. The LDAOR has the lowest overhead since the messages are only delivered to appropriate nodes, thus avoiding additional transmissions of messages. Hence, buffer of nodes are less occupied and traffic loss in buffers reduces in LDAOR as shown in both Fig. 6c and Fig. 7c. As shown in Fig. 6c and Fig. 7c, by increasing traf-

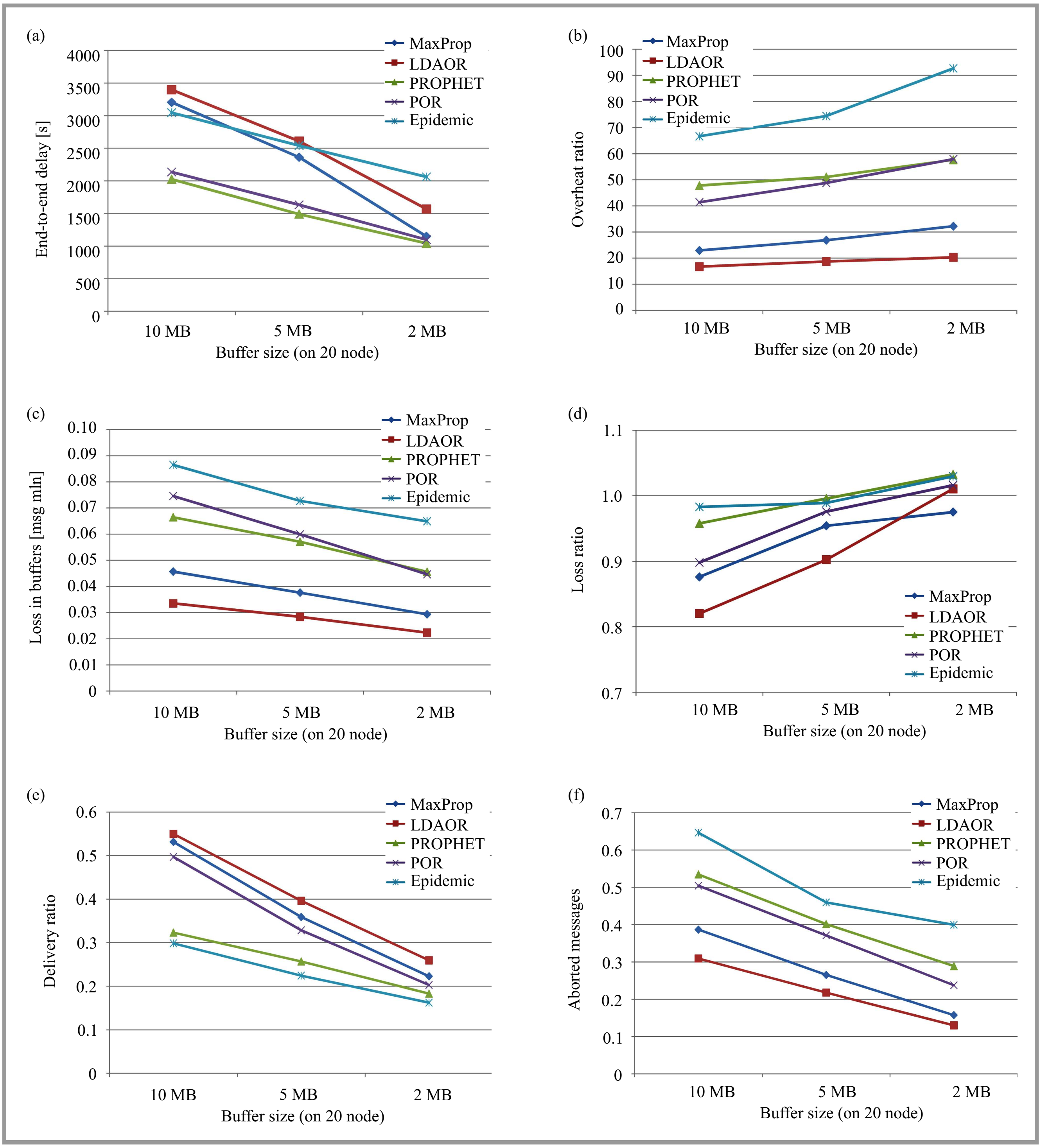

Fig. 8. Traffic load of 1 packets/Uniform(5, 15) sec under different buffer sizes: (a) end-to-end delay, (b) overhead ratio, (c) loss in buffers, (d) loss ratio, (e) delivery ratio, (f) aborted messages.

fic load, loss in buffers increases compared to Fig. 5c. This is because those messages that cannot be delivered should still remain in buffers of nodes. On the other hand, new traffic is always generated. Hence, buffers will overflow, thus resulting in dropping more messages.

As aforementioned, message delivery rate reduces by increasing traffic load (see Fig. 6d and Fig. 7d). As a result, in order to fully utilize the maximum bandwidth, selecting the best forwarder node for messages and their prioritization finds importance. As it can be observed, the LDAOR has the highest message delivery rate compared to other protocols, as shown in Fig. 6d and Fig. 7d.

By limited number of transmitted messages compared to more generated messages at high traffic load, the number of aborted messages goes down (compare Fig. 5e with Fig. 6e and Fig. 7e). The LDAOR still experiences the least number of aborted messages compared to other routing protocols even at high traffic load, as shown in Fig. 6e and Fig. 7e.

Figure 8 shows performance of network under different buffer sizes in a VANET with 20 nodes. As it can be observed, reducing the buffer size reduces delivery ratio due to high limitation on buffer size. Notice when a buffer is full, messages should be removed from the buffer. As a result, a transmitted message may be removed from a buffer before being delivered to its destination. This issue increases overhead and decreases delivery ratio. When the buffer size reduces, the number of relayed messages reduces as well since small number of messages can be saved in buffers. Note the ratio of the number of dropped messages over the number of relayed message increases, thus increasing loss ratio (as shown in Fig. 8d). Since end-to-end delay is computed based on successfully delivered messages to their destinations, end-to-end delay also decreases because of decreasing the delivery ratio.

5. Conclusion

The LDAOR method has been proposed for opportunistic VANET in order to improve the performance of routing. The idea behind this approach is to consider physical location and direction of vehicles for choosing the best forwarder node among multiple neighbor nodes. It has been shown that LDAOR can provide better performances compared to other conventional routing protocols even when resources such as buffers are limited and traffic density is high. The LDAOR reduces traffic loss, aborted messages, and overhead ratio. On the other hand, it increases the probability of successful message delivery. The LDAOR provides smaller end-to-end delay at low traffic loads as well. Although the delivery ratio and overhead in LDAOR is not significantly different from MaxProp, but the differences between LDAOR and MaxProp in terms of end-to-end delay, loss in buffers and aborted messages are considerable. The complexity of LDAOR depends on the number of neighbor nodes in each contact. However, the number of neighbor nodes in opportunistic networks is not high in practice.

Acknowledgements

We would like to thank Mr. Ari Keränen for his assistance in using the ONE simulator.

References

[1] K. C. Lee and M. Gerla, "Opportunistic vehicular routing", in *Proc. Eur. Wirel. Conf. EW 2010*, Lucca, Italy, 2010, pp. 873–880.

[2] A. Casteigts, A. Nayak, and I. Stojmenovic, "Communication protocols for vehicular ad hoc networks", *Wirel. Commun. Mob. Comput.*, vol. 11, no. 5, pp. 567–582, 2011.

[3] R. C. Shah, S. Wietholter, J. Rabaey, and A. Wolisz, "When does opportunistic routing make sense?", in *Proc. 3rd IEEE Int. Conf. Pervasive Comput. Commun. Worksh. PerCom 2005*, Kauai Island, HI, USA, 2005, pp. 350–356.

[4] Z. Zhang and Q. Zhang, "Delay/disruption tolerant mobile ad hoc networks: latest developments", *Wirel. Commun. Mob. Comput.*, vol. 7, no. 10, pp. 1219–1232, 2007.

[5] W. Chen, R. K. Guha, T. J. Kwon, J. Lee, and Y.-Y. Hsu, "A survey and challenges in routing and data dissemination in vehicular ad hoc networks", *Wirel. Commun. Mob. Comput.*, vol. 11, no. 7, pp. 787–795, 2011.

[6] X. Zhao, "An adaptive approach for optimized opportunistic routing over delay tolerant mobile ad hoc networks", Ph.D. thesis, Computer Science Department, Rhodes University, Grahamstown, South Africa, 2007.

[7] R. J. D'Souza and J. Jose, "Routing approaches in delay tolerant networks: A survey", *Int. J. Comp. Appl.*, vol. 1, no. 17, pp. 8–14, 2010.

[8] A. Vahdat and D. Becker, "Epidemic routing for partially connected ad hoc networks", Tech. Rep. CS-200006, Duke University, 2000.

[9] J. Burgess, B. Gallagher, D. Jensen, and B. N. Levine, "MaxProp: routing for vehicle-based disruption tolerant networks", in *Proc. 25th IEEE Int. Conf. Comp. Commun. INFOCOM 2006*, Barcelona, Spain, 2006, pp. 1–11.

[10] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Single-copy routing in intermittently connected mobile networks", *IEEE/ACM Trans. on Netw.*, vol. 16, no. 1, pp. 63–76, 2008.

[11] T. Spyropoulos, K. Psounis, and C. Raghavendra, "Spray and wait: an efficient routing scheme for intermittently connected mobile networks", in *Proc. ACM SIGCOMM Worksh. Delay-tolerant Netw. WDTN'05*, Philadelphia, PA, USA, 2005, pp. 252–259.

[12] S. C. Nelson, M. Bakht, R. Kravets, and A. F. Harris, "Encounter based routing in DTNs", in *Proc. 28th Conf. Comp. Commun. IEEE INFOCOM 2009*, Rio de Janeiro, Brazil, 2009, pp. 846– 854.

[13] H. Kang and D. Kim, "Vector routing for delay tolerant networks", in *Proc. 68th IEEE Veh. Technol. Conf. VTC 2008-Fall*, Calgary, Canada, 2008, pp. 122–135.

[14] E. Bulut, Z. Wang, and B. K. Szymański, "Cost-effective multi period spraying for routing in delay-tolerant networks", *IEEE/ACM Trans. on Netw*., vol. 18, no. 5, pp. 1530–1543, 2010.

[15] K. A. Harras and K. C. Almeroth, "Controlled flooding in disconnected sparse mobile networks", *Wirel. Commun. Mob. Comput.*, vol. 9, no. 1, pp. 21–33, 2009.

[16] L. M. Kiah, L. K. Qabajeh, and M. M. Qabajeh, Unicast position-based routing protocols for ad-hoc networks", *Acta Polytechnica Hungarica*, vol. 7, no. 5, pp. 19–46, 2010.

[17] H. Takagi and L. Kleinrock, "Optimal transmission ranges for randomly distributed packet radio terminals", *IEEE Trans. on Commun.*, vol. 32, no. 3, pp. 246–257, 1984.

[18] X. Li, M. Li, W Shu, and M. Y. Wu, "Packet-oriented routing in delay-tolerant vehicular sensor networks", *J. Inform. Sci. Engin. (JISE)*, vol. 25, no. 6, pp. 1803–1817, 2009.

[19] P.-C. Cheng, K. C. Lee, M. Gerla, and J. Härri, "GeoDTN+Nav: Geographic DTN routing with navigator prediction for urban vehicular environments", *Mob. Netw. Appl.*, vol. 15, no. 1, pp. 61–82, 2010.

[20] I. Jang, W. Choi, and H. Lim, "An opportunistic forwarding protocol with relay acknowledgment for vehicular ad hoc networks", *Wirel. Commun. Mob. Comput.*, vol. 11, no. 7, pp. 939–953, 2011.

[21] A. Lindgren, A. Doria, and O. Schelen, "Probabilistic routing in intermittently connected networks", *ACM SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 7, no. 3, pp. 19–20, 2003.

[22] D. Yu and Y.-B. Ko, "FFRDV: Fastest-ferry routing in DTN-enabled vehicular ad hoc networks", in *Proc. 11th Int. Conf. Adv. Commun. Technol. ICACT 2009*, Phoenix Park, Korea, 2009, vol. 2, pp. 1410–1414.

[23] Y. Ding, C. Wang, and L. Xiao, "A static-node assisted adaptive routing protocol in vehicular networks", in *Proc. 4th ACM Int. Worksh. Veh. Ad Hoc Netw. VANET'07*, Montreal, Quebec, Canada, 2007, pp. 59–68.

[24] I. Leontiadis and C. Mascolo, "GeOpps: Geographical opportunistic routing for vehicular networks", in *Proc. IEEE Int. Symp. World of Wirel., Mob. Multimed. Netw. WoWMoM 2007*, Helsinki, Finland, 2007.

[25] C. Liu and J. Wu, "An optimal probabilistically forwarding protocol in delay tolerant networks", in *Proc. 10th ACM Int. Symp. Mob. Ad Hoc Netw. Comput. ACM MobiHoc 2009*, New Orleans, Louisiana, USA, 2009, pp. 105–114.

[26] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Spray and focus: effcient mobility-assisted routing for heterogeneous and correlated mobility", in *Proc. 5th IEEE Int. Conf. Pervasive Comput. Commun. Worksh. PerCom 2007*, White Plains, New York, USA, 2007.

[27] Routing in delay-tolerant networking [Online]. Available: http://en.wikipedia.org/wiki/Routing_in_delay-tolerant_networking

[28] H. Fubler, M. Mauve, H. Hartenstein, D. Vollmer, and M. Kasemann, "Location-based routing for vehicular ad-hoc networks", *ACM SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 7, no. 1, pp. 47–49, 2003.

[29] M. Jerbi, S.-M. Senouci, T. Rasheed, and Y. M. Ghamri-Doudane, "Towards efficient geographic routing in urban vehicular networks", *IEEE Trans. Veh. Technol.*, vol. 58, no. 9, pp. 5048–5059, 2009.

[30] F. Li and Y. Wang, "Routing in vehicular ad hoc networks: A survey", *IEEE Veh. Technol. Mag.*, vol. 2, no. 2, pp. 12–22, 2007.

[31] J. M. Soares, M. Franceschinis, R. M. Rocha, W. Zhang, and M. A. Spirito, "Opportunistic data collection in sparse wireless sensor networks", *EURASIP J. Wirel. Commun. Netw*., vol. 2011, pp. 6:1–6:20, 2011.

[32] G. Owen and M. Adda, "SOLS: Self organizing distributed location server for wireless ad hoc networks", *Int. J. Comp. Netw. & Commun. (IJCNC)*, vol. 1, no. 1, pp. 17–30, 2009.

[33] S. Farrell *et al*., "Report on an arctic summer DTN 2010 trial", Tech. rep., Trinity College, Dublin, Ireland, 2011 (draft 2011-05-18 Work-in-progress) [Online]. Available: http://dtn.dsg.cs.tcd.ie/n4c-summer10/summer10.pdf

[34] J. A. B. Link, C. Wollgarten, S. Schupp, and K. Wehrle, "Perfect difference sets for neighbor discovery energy efficient and fair", in *Proc. 3rd ACM Extreme Conf. on Commun. ExtremeCom 2011*, Manaus, Brazil, 2011.

[35] Opportunistic Network Environment (ONE) homepage, ver. 1.4.1 [Online]. Available: http://www.netlab.tkk.?/%7Ejo/dtn/ (accessed June 2011).

[36] V. Soares, J. Rodrigues, P. Salvador, and A. Nogueira, "Improvement of messages delivery time on vehicular delay-tolerant networks", in *Proc. Int. Worksh. Next Gener. Wirel. Mob. Netw. NGWMN 2009*, Vienna, Austria, 2009.

[37] A. Keränen, T. Kärkkäinen, and J. Ott, "Simulating mobility and DTNs with the ONE", *J. Commun.*, vol. 5, no. 2, pp. 92–105, 2010.

[38] V. Soares, F. Farahmand, and J. R. Rodrigue, "Improving vehicular delay-tolerant network performance with relay nodes", in *Proc. 5th Euro-Ngi Conf. Next Gener. Internet Netw. NGI 2009*, Aveiro, Portugal, 2009.

[39] F. Alnajjar and T. Saadawi, "Performance analysis of routing protocols in delay/disruption tolerant mobile ad hoc networks", in *Proc. 10th WSEAS Int. Conf. on Elec., Hardw., Wirel. Opt. Commun. EHAC'11, and 10th WSEAS Int. Conf. on Sig. Process., Robot. and Autom. ISPRA'11, and 3rd WSEAS Int. Conf. Nanotechnol. NANOTECHNOLOGY'11*, Cambridge, UK, 2011, pp. 407–417.

Marziyeh Barootkar received her B.S. degree in Software Engineering in 2008 from Zanjan University, Zanjan, Iran, and M.Sc. degree in Information Technology from Sahand University of Technology in 2011. Her research interests include wireless, MANET and VANET networks and performance evaluation.

E-mail: m_barootkar@sut.ac.ir
Computer Networks Research Lab
Electrical Engineering Technologies Research Center Sahand University of Technology
Tabriz, Iran

Akbar Ghaffarpour Rahbar received the B.Sc. and M.Sc. degrees in computer hardware and computer architecture from the Iran University of Science and Technology, Tehran, Iran, in 1992 and 1995, respectively, and the Ph.D. degree in computer science from the University of Ottawa, Ottawa, Canada, in 2006. He is currently a Professor with the Electrical Engineering Department, Sahand University of Technology, Sahand New Town, Tabriz, Iran. He is the director of the Computer Networks Research Laboratory, Sahand University. Dr. Rahbar is a senior member of the IEEE. He is currently on the Editorial Board of the Wiley Transactions on Emerging Telecommunications Technologies Journal and the Journal of Convergence Information Technology. He is editor-in-chief of Journal of Nonlinear Systems in Electrical Engineering. His current research interests include optical networks, optical packet switching, scheduling, PON, IPTV, network modeling, analysis and performance evaluation, the results of which can be found in over 110 technical papers.

E-mail: ghaffarpour@sut.ac.ir
Computer Networks Research Lab
Electrical Engineering Technologies Research Center
Sahand University of Technology
Tabriz, Iran

Masoud Sabaei received his B.Sc. degree from Isfahan University of Technology, Isfahan, Iran, and his M.Sc. and Ph.D. form Amirkabir University of Technology (Tehran Polytechnic), Tehran, Iran, all in the field of Computer Engineering in 1992, 1995 and 2000, respectively. Dr. Sabaei has been professor of Computer Engineering Department, Amirkabir University of Technology (Tehran Polytechnic), Tehran, Iran since 2002. His research interests are software defined networking, Internet of Things, wireless networks and telecommunication network management.

E-mail: sabaei@aut.ac.ir
Computer Engineering and Information
Technology Department
Amirkabir University of Technology
Tehran, Iran