

# Trading Optimality for Scalability in Large-scale Opportunistic Routing

Yanhua Li, *Student Member, IEEE*, Abedelaziz Mohaisen, *Member, IEEE*, and Zhi-Li Zhang, *Fellow, IEEE*

**Abstract**—Opportunistic routing utilizes the broadcast nature of wireless networks, significantly promoting the unicast throughput. Many variations of opportunistic routing designs have been proposed, though all of the current designs consistently rely on the whole topology information to construct forwarder lists and process data forwarding, which indeed restricts the application in large-scale wireless networks—where collecting global optimal information is very costly. In this paper, we propose the localized opportunistic routing (LOR) protocol, which utilizes the distributed minimum transmission selection (MTS-B) algorithm to partition the topology into several nested close-node-sets (CNS's) using local information. LOR can locally realize the optimal opportunistic routing for large-scale wireless network with low control overhead cost. Since it does not use global topology information, LOR highlights an interesting trade-off between the global optimality of the used forwarder lists and scalability inferred from the incurred overhead. Extensive simulation results show that LOR dramatically improves the performances over ExOR and MORE, two well-known designs from the literature, in terms of control overhead, end-to-end delay, and throughputs. It also exhibits promising performance in VANETs.

**Index Terms**—Opportunistic routing, graph partitioning, local information, distributed routing.

## I. INTRODUCTION

The opportunistic routing is a new design trend of wireless routing protocols. Unlike traditional routing — such as DSR [15] and AODV [30] in which source takes one “best” path to the destination, in the opportunistic routing protocol the source takes multiple opportunistic paths to forward packets to the destination. The path a packet takes depends on which forwarders happen to receive it, and is thus non-deterministic. Also, if a certain wireless forwarder fails or moves out of the radio range during the transmission, other possible paths may take over the transmissions. As a result, opportunistic routing can better cope with the *lossy, unreliable, and varying* link qualities that are typical of wireless networks.

ExOR [3] is the primary opportunistic routing protocol in the literature. Using ETX [5] as a routing metric, ExOR constructs forwarder list for each source-destination pair, in which case all forwarders are prioritized. The source node

broadcasts data packets, which carry the forwarder list information in their headers. Many neighbor forwarders of source may hear the transmission but only the highest priority forwarder that received the packet would forward it and all of other forwarders in the forwarder list would have to wait for their higher priority nodes’ transmissions, so that the packet always be forwarded by the node nearest to the destination.

More recently, many opportunistic strategies have been proposed to address various problems. For examples, MORE [4] presents the observation that during the ExOR data forwarding process, the nodes in the forwarder list transmit in strict order, in such a way as to reduce the duplicate transmissions. These forwarders cannot transmit at the same time, even though they are not within the same radio range. This would result in high end-to-end latency, especially when the source and the destination are multiple radio ranges away, which leads to what is called “spatial reuse” problem. The spatial reuse is gracefully solved in MORE [4] by exploiting linear network coding technique. In [19] we proposed the minimum transmission selection (MTS) scheme, which can choose the optimal forwarder list for source-destination pair in a centralized (Dijkstra’s-like) manner. In [20], we made the first attempt to address the problem of applying opportunistic routing in large-scale wireless network, and proposed the graph partition based local scheduling (GPLS) protocol — which utilizes the spectral graph partitioning algorithm to decompose wireless topology into subgraphs and to realize local opportunistic routing. GPLS drastically reduces the data plane overheads.

However, to the best of our knowledge, all existing opportunistic routing protocols have to rely on global information of the whole wireless topology to compute the forwarder list for each node pair. This is, each node has to periodically build the network graph with all link qualities among the whole topology. In large-scale networks, it would be a tedious task.

In this paper, by building on top of our previous work in [26], we first propose the distributed minimum transmission selection MTS-B scheme, which realizes optimal forwarder list selection using distributed (Bellman-Ford-like) algorithm. The main reason we use the Bellman-Ford algorithm in this work is that it can be implemented in a distributed manner as in, for example, the routing information protocol (RIP) [12], unlike the Dijkstra’s algorithm, which is centralized by nature. Accordingly, the base algorithm in [26] and our extension in this paper are not meant to handle negative weights. When using MTS-B, we analyze the properties of the ranking matrix of the wireless topology and show that every topology could be partitioned into several nested node sets called *close-node-set* (CNS). If the source and the destination are within the

Manuscript received March 28, 2012; revised June 29, 2012, October 12, 2012, and December 7, 2012; accepted December 13, 2012; approved by IEEE Transactions on Vehicular Technologies editor G. Mao. The work is supported in part by the NSF grants CNS-0905037, CNS-1017647 and the DTRA Grant HDTRA1-09-1-0050. An earlier version of this work appeared in the Proceedings of IEEE GLOBECOM 10’ [26], December 2010. Copyright (c) 2012 IEEE. Personal use of this material is permitted. However, permission to use this material for any other purposes must be obtained from the IEEE by sending a request to pubs-permissions@ieee.org.

Yanhua Li and Zhi-Li Zhang are with the Department of Computer Science and Engineering, University of Minnesota, Minneapolis, MN 55455 USA. Abedelaziz Mohaisen is with Verisign Labs, Reston, VA 20190 USA (e-mail: yanhua@cs.umn.edu; amohaisen@verisign.com; zhzhzhang@cs.umn.edu).

same CNS, their optimal forwarder list is also within the same CNS. Otherwise, the source needs to know how to get to the CNS which the destination belongs to — instead of the exact optimal forwarder list to the destination node. The CNS’s form an overlay network on top of wireless topology, thus the local opportunistic paths become transparent to the node pair. Based on the observation made on the CNS, we propose the localized opportunistic routing (LOR) protocol, which can locally realize optimal opportunistic routing in large-scale wireless network, by partitioning the topology into several CNS’s using local information. To this end, we summarize the three main contributions of this paper as follows:

- First, we propose the distributed minimum transmission selection (MTS-B) algorithm, which can select optimal forwarder list for node pair in a *distributed* fashion. It is more suitable for large-scale wireless networks, where collecting global topology information is infeasible.
- We propose the localized opportunistic routing (LOR), which is the first work for designing a scalable opportunistic routing requiring only local information. In LOR, each node employs just sufficient local information to realize the opportunistic routing. LOR drastically reduces the overheads in both control plane and data plane.
- We present a graph partitioning method to decompose a large-scale wireless network into small autonomous sub-topologies, using local information. Each sub-topology could realize local optimal opportunistic routing by itself.

**Organization.** We discuss the related work in Section II, and the problem statement Section III. Then, the distributed minimum transmission selection (MTS) algorithm is introduced in Section IV. In Section V, the localized opportunistic routing protocol is presented. LOR is evaluated in Section VI. The paper is concluded in Section VII.

## II. RELATED WORK

Dubois-Ferriere *et al.* [8] introduced a specific cost function defined with respect to a set of candidate forwarders, and proposed the least-cost opportunistic routing (LCOR) algorithm to identify the best candidate set that minimizes the said cost function. Due to its exponential time complexity, heuristic policies have to be incorporated in LCOR. Zeng *et al.* [36] studied end-to-end throughput of opportunistic routing in *multi-rate* networks using a linear programming framework.

Other protocols with various opportunistic routing flavors are proposed. CodeOR [27] used networking coding in opportunistic routing, and realized transmitting a windows of multiple coded batches simultaneously. Zhong *et al.* [38] proposed the expected anycast transmissions as a metric to capture the expected number of transmissions needed to opportunistically deliver a packet between two nodes, and resorted to heuristic algorithms for computing a set of candidate forwarders.

Many forwarding algorithms [9], [16], [31], [6], [7] try to reduce the control overheads and increase the success of transmission rate. FRESH [7] relied on node’s last encounter time with the destination to make a forwarding decision. Greedy [9] utilized the greedy-total contact rate of nodes to select transmission candidates. Acer *et al.* [2] proposed weak state routing mechanism, with which wireless nodes transmit

packets in a “biased random walks” manner. In EASE [11], every node maintains an encounter history consisting of the location and the time of its last encounter with other nodes to efficiently find routes to the destination using this history. However, these works are designed based on the traditional routing framework which relies on one single best path, slightly differing from our approach.

In [20] we proposed GPLS, which utilizes spectral clustering algorithm to partition the topology into subgraphs, and realizes opportunistic routing locally to improve the scalability. However, the spectral clustering method has to rely on global topology information, and it does not take into consideration the key properties of opportunistic routing, thus can not achieve optimal performance in each subtopology. In this paper, we design localized opportunistic routing (LOR) protocol, which partitions the topology using the close-node-set (CNS) theory. We prove that LOR can realize local optimal performance for opportunistic routing in large-scale networks.

Closely related to the CNS formation algorithm we propose in this work is the clustering in mobile ad-hoc networks (MANETs) and vehicular ad-hoc networks (VANETs) [32], [35], [18]; a nice survey on algorithms in the former type of networks is in [35] while clustering algorithms in the latter networks are surveyed in [18]. In short, unlike these clustering algorithms, our work avoids reliance on global topological information, does not consider or rely on a single optimal path in routing, and is fully opportunistic.

## III. PROBLEM DEFINITION

Existing opportunistic routing schemes — including ExOR, MORE, and MTS — can not be applied in large-scale wireless network very well because all of those schemes are designed as centralized algorithms and have to rely on the whole topology information, which is costly to realize in practice especially when the topology is highly dynamic and time-varying. These give the raise to the problem, whether we could realize opportunistic routing in a *distributed manner with only sufficient, yet limited, local information*. We in fact solve this problem in two steps in this paper. First, we design a distributed opportunistic routing scheme that can produce the global optimal solution. Then, we propose a localized routing scheme that only uses local information to partition the topology into subgraphs and to realize local optimal opportunistic routing, in such a way that dramatically reduces the control overheads as well as maintains the locally minimized transmission delay. In this section, we will use a simple example to show the sufficiency of local information for realizing opportunistic routing and give a brief overview of our main idea.

Fig. 1 shows an example wireless topology, with six wireless nodes and edge weights representing the link quality, i.e., the success rates of transmissions over those links. Using MTS, every node  $v_i$  in the topology needs to exchange Hello/Probe

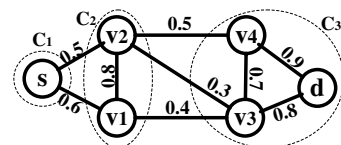


Fig. 1. Example topology

messages with its neighbors to collect the whole topology information. Then using the obtained whole topology information, each node can compute the global optimal forwarder lists to any other nodes, e.g., the source node  $s$  has the optimal forwarder list  $\{s, v_1, v_2, v_3, v_4, d\}$  to destination  $d$ . However, in fact, local information is sufficient for the source to route data to the destination. This can be done by dividing the topology  $G$  into 3 separate node sets, as  $C_1 = \{s\}$ ,  $C_2 = \{v_1, v_2\}$ ,  $C_3 = \{v_3, v_4, d\}$ —notice that this is one such example of the topology grouping, where other grouping can serve the same purpose. It is easy to check that if a pair of nodes is within the same node set  $C_i$ , their local optimal forwarder lists in  $C_i$  are exactly the same as their global optimal forwarder lists in  $G$ . If a source is not in the same subgraph as the destination, it just needs to know the node set path to the destination instead of the exact global forwarder list. These can be verified by Table I, which shows the optimal forwarder lists from all possible nodes to the nodes in set  $C_3$ . One could check that every  $C_i$  has this property. For the pair  $(s, d)$ ,  $s$  just needs to know its set path as  $C_1 \rightarrow C_2 \rightarrow C_3$  to forward the packets to  $d$ . When packets reach set  $C_2$ , nodes in  $C_2$  could easily find local (also global) opportunistic list, i.e.  $\{v_1, v_2\}$ , to its next hop set  $C_3$ . Once the packet reaches  $C_3$ , it can locally find optimal opportunistic list to the destination  $d$  as  $\{v_3, v_4, d\}$ . Clearly, merging those local forwarder lists together, the packets take the global optimal forwarder to  $d$ , as MTS does. Hence, local information is sufficient for node pairs to find their global optimal forwarder lists.

Notice that, as shown in Table I, whether an intermediate node  $i$  should be taken into the forwarder list for node pair  $(a, b)$  depends on whether  $N(i, b) < N(a, b)$ , and whether node  $i$  should be taken into the forwarder list for node pair  $(b, a)$  depends on whether  $N(i, a) < N(b, a)$ . These two conditions are independent.

TABLE I  
GLOBAL AND LOCAL FORWARDER LISTS FOR NODE SET  $C_3$  IN FIG1

Node-pair	Global list = Local list
$(v_3, v_4)$	$\{v_3, d, v_4\}$
$(v_3, d)$	$\{v_3, v_4, d\}$
$(v_4, v_3)$	$\{v_4, d, v_3\}$
$(v_4, d)$	$\{v_4, d\}$
$(d, v_3)$	$\{d, v_3\}$
$(d, v_4)$	$\{d, v_4\}$

To this end, in the rest of the paper we answer the two questions below, and formally formulate these insights and realize the scalable opportunistic routing in large-scale wireless network with just sufficient local information at each node.

- How to design a partitioning algorithm for wireless topology, which realizes global optimal forwarder lists within each local sub-topology with only local information?
- How to utilize the local partitioning information to design an opportunistic routing protocol that reduces the control overhead as well as the expected transmission delay?

#### IV. DISTRIBUTED FORWARDER LIST SELECTION SCHEME

In [19], we presented the minimum transmission selection (MTS) algorithm which can select global optimal forwarder list for opportunistic routing with minimized expected transmission delay. MTS has the centralized structure analogous

to Dijkstra's algorithm, with the expected number of opportunistic transmissions as a metric. In this section, we will first propose the distributed minimum transmission algorithm (MTS-B), which selects the global optimal forwarder list in a distributed fashion. Then, we will prove its optimality. In the next section, we will show how the MTS-B algorithm can be used for partitioning the wireless topology and realize opportunistic routing with local information.

#### A. Notations and Opportunistic Routing in Wireless Networks

Given a topology denoted as a graph  $G = (V, E)$  with  $|V| = N$  wireless nodes, let  $FL^s(\cdot)$  denote the forwarder lists vector of node  $s$  where the entry  $FL^s(v_i)$  is current forwarder lists from  $s$  to  $v_i$ . Let  $N^s(\cdot)$  be the number of transmissions vector, where  $N^s(v_i)$  is the number of transmissions from  $s$  to  $v_i$  when using the current forwarder list  $FL^s(v_i)$ .

TABLE II  
TERMINOLOGY

Term	Definition
$N^s(d)$	Expected number of transmissions from $s$ to $d$
$FL^s(d)$	Forwarder list for node pair $(s, d)$
$N(\cdot)$	Estimated number of transmissions
$B_s$	Neighbor set of node $s$
$B_s(i)$	The $i$ -th neighbor of node $s$
$N_0^s(d)$	Initial number of transmissions from $s$ to $d$
$FL_0^s(d)$	Initial forwarder list for node pair $(s, d)$

By exchanging the Probe/Hello messages, each node  $s$  can estimate its one-hop link qualities  $Pr(s, v_i)$  to all of its neighbors, which captures the probability for a packet to be successfully transmitted from  $s$  to  $v_i$  in each attempt. With the one-hop link quality information, and assuming that the underlying communication wireless channel is time uncorrelated, each node  $s$  can initiate  $FL^s(\cdot)$  and  $N^s(\cdot)$  by

$$N^s(v) := \begin{cases} 1/Pr(s, v) & \text{if } v \in B_s \\ \infty & \text{if } v \notin B_s \end{cases}, \quad (1)$$

$$FL^s(v) := \begin{cases} \{s, v\} & \text{if } v \in B_s \\ \emptyset & \text{if } v \notin B_s \end{cases}. \quad (2)$$

Then, each node  $s$  exchanges the local information  $FL^s(\cdot)$  and  $N^s(\cdot)$  with neighbors, by piggybacking them in the Probe/Hello messages. Upon receiving a forwarder list information from its neighbor, node  $s$  updates its own information to reflect any changes. Then,  $s$  broadcasts its updated information, so that other nodes can update further.

For estimating one-hop link qualities we use the following. Every node sends periodically a hello messages. E.g., 100 times in 1 minute. Once the neighboring node receives the message it needs to respond to the message with an acknowledgement. The sender needs to calculate the quality of the one-hop link from the number of responses. In particular, an estimate of link quality is the number of successfully received acknowledgements normalized by the number of total messages sent.

Notice that the aforementioned method for estimating the one-hop link qualities requires acknowledgement messages to be error-free, since their accurate transmission is required for estimating the link state. This can be further ensured

using proper channel coding. Furthermore, given that the number of times a hello message is transmitted over a period of time is fixed and known in advance, the overhead of acknowledging all hello messages can be summarized by a single acknowledgement from the receiver with the number of successfully received hello messages. With the proper channel coding in place, the acknowledgement is delivered accurately to the sender estimating the link state.

Algorithm 1 shows an algorithmic description of the distributed MTS-B. The initialization process is in Algorithm 2. Each node  $s$  initiates its local information using eq. (1) and eq. (2) in Algorithm 2. Then,  $s$  sends these one-hop information out (**Line 9**). In each iteration (processed in Algorithm 1), every node updates its information upon receiving an update message from its neighbor. **Lines 2-13** are the outer loop, which continues until no update message is received or its own forwarder list information gets stable. At **Lines 3-12**,  $s$  updates its own information for all possible  $N - 1$  destinations. For each source-destination pair  $(s, d)$ ,  $s$  orders its own neighbors by the current numbers of transmissions to  $d$  (**Line 4**). Two intermediate variables  $N^s$  and  $FL^s$  are initiated at **Line 5**. For each neighbor  $B_s(i)$  (where  $i = 1, \dots, M$ ), if  $N^s(d) > N^{B_s(i)}(d)$  holds — which means  $B_s(i)$  has less expected number of transmissions to  $d$  than the source node  $s$ , node  $B_s(i)$  should be introduced in the forwarder list of node  $s$ . Note that at this stage, node  $s$  needs to decide whether to substitute the existing forwarder list to  $d$  by  $\{s, FL^{B_s(i)}(d)\}$  or to merge them together as **merge** $\{FL^s(d), FL^{B_s(i)}(d)\}$ , and this is determined by which choice reaches a lower number of transmissions (See **Lines 8-12**). For that last step, the node set **merge** $\{FL^1(d), FL^2(d)\}$  includes each node  $v \in FL^1(d) \cup FL^2(d)$ , decreasingly ordered by the expected number of transmissions  $N^v(d)$ . This condition checking is necessary because sometimes the information from the best neighbors is received later than other neighbors due to the non-synchronized packet arrivals.

---

**Algorithm 1** MTS-B Algorithm(list vertices, list edges, vertex  $s$ )

---

```

1: Initiate( $s$ );
2: while (Receive  $N^u(\cdot)$  from  $u$ ) && ( $N^s(\cdot)$  is not stable) do
3:   for each vertex  $d \in V/\{s\}$  do
4:      $s$  orders its  $M$  neighbors by  $N^{B_s(1)}(d) \leq \dots \leq N^{B_s(M)}(d)$ .
5:      $N^s := N_0^s(d)$  and  $FL^s := FL_0^s(d)$ ;
6:     for each neighbor  $B_s(i)$ ,  $i := 1, \dots, M$  do
7:       if  $N^s > N^{B_s(i)}(d)$  then
8:         if  $\mathcal{N}(\text{merge}\{FL^{B_s(i)}(d), FL^s\}) \leq \mathcal{N}(\{s, FL^{B_s(i)}(d)\})$ 
9:           then
10:             $N^s := \mathcal{N}(\text{merge}\{FL^{B_s(i)}(d), FL^s\})$ ;
11:             $FL^s := \text{merge}\{FL^{B_s(i)}(d), FL^s\}$ ;
12:           else
13:             $N^s := \mathcal{N}(\{s, FL^{B_s(i)}(d)\})$ ,  $FL^s := \{s, FL^{B_s(i)}(d)\}$ ;
13:   Send  $FL^s(\cdot)$  and  $N^s(\cdot)$  out.
```

---

Notice that the term stability is defined so that the difference of  $N(s, d)$  between the  $i$ -th and  $(i + m)$ -th steps is small enough. Formally, given a small  $\delta > 0$ , and a large integer  $m$ , stability means that for steps after a certain step  $i$ ,  $N(s, d)$  at step  $i$  minus  $N(s, d)$  at step  $i + m$  is less than or equal  $\delta$ .

Moreover, from algorithm 1, each node updates its forwarder lists to all other nodes in the topology. Given a topology with  $N$  nodes, every node needs at the worst case  $N^2$  steps to update its forwarder lists to be optimal. Hence,

---

**Algorithm 2** initiate(vertex  $s$ )

---

```

1: for each vertex  $v \in E/\{s\}$  do
2:   if  $P_r(s, v) > 0$  then
3:      $N^s(v) := 1/P_r(s, v)$ ;  $FL^s(v) := \{s, v\}$ 
4:   else
5:      $N^s(v) := \infty$ ;  $FL^s(v) := \emptyset$ ;
6:   for each neighbor  $B_s(j)$  of  $s$  do
7:      $N^{B_s(j)}(v) = \infty$ 
8:    $N_0^s(\cdot) := N^s(\cdot)$  and  $FL_0^s(\cdot) := FL^s(\cdot)$ ;
9: Send  $FL^s(\cdot)$  and  $N^s(\cdot)$  out;
```

---

the time complexity of Algorithm 1 is  $\mathcal{O}(N^2)$ , with which the network link qualities need to be relatively stable so as to maintain a low convergence rate for achieving optimal network-wide forwarding.

### B. Optimality of MTS-B Algorithm

The optimality of the centralized MTS algorithm is proved in our previous work in [19]. Here, we prove the optimality of MTS-B and show that it can always converge to the global optimal forwarder list for each source-destination pair.

First of all, we provide some properties of the optimal forwarder list for a source-destination pair in opportunistic routing. Those properties state the essential insights of the forwarder list selection process, and lead us to prove the optimality of MTS-B algorithm. Recall that given a forwarder list between  $s$  and  $d$ , denoted as  $FL = \{s, v_m, \dots, v_1, d\}$ , the expected number of transmissions  $\mathcal{N}(FL)$  can be computed using the formulation in [19] as

$$\mathcal{N}(FL) = \frac{1 + \sum_{k=1}^m x_k \prod_{j=1}^{k-1} (1 - x_j)(1 - z) \mathcal{N}_{v_k, \dots, v_1, d}}{1 - \prod_{k=1}^m (1 - x_k)(1 - z)}, \quad (3)$$

where  $x_k = Pr(s, v_k)$ ,  $z = Pr(s, d)$ . Then, the optimal forwarder list  $FL^s(d)$  is defined as the one with the least number of transmissions between  $s$  and  $d$ . Lemma 1 specifies the condition under which the optimal forwarder list is computed.

**LEMMA 1.** *Each node  $s$  can compute its optimal forwarder list  $FL^s(d)$  to  $d$  if it has the information of the optimal forwarder lists and the corresponding expected number of transmissions from all of its higher priority neighbors  $v_i$ 's, where  $N^{v_i}(d) \leq N^s(d)$ , and  $v_i \in B_s(d)$ .*

*Proof:* Given a source-destination pair  $(s, d)$ , suppose its optimal forwarder list is  $\{s, v_{n-1}, \dots, v_1, d\}$ ,  $1 \leq n \leq N - 1$  within which  $m \leq n - 1$  forwarders are  $s$ ' neighbors, denoted by  $u_m, \dots, u_1$ , with order  $N^{u_m}(d) \geq \dots \geq N^{u_1}(d)$ . Note that when  $n = 1$ , the forwarder list is in form of  $\{s, d\}$ , with no intermediate forwarders included.

If node  $s$  has the optimal forwarder lists and the corresponding expected number of transmissions from all higher priority neighbors  $u_i$ 's ( $1 \leq i \leq m$ ), based on MTS algorithm [19], the optimal forwarder list of  $s$  is obtained from  $FL^s(d) = \text{merge}_{i=1}^m (FL^{u_i}(d))$ , and the expected number of transmissions is computed by Eq. (3). The optimality follows the proof of optimality for MTS algorithm. ■

Now we are in the position to prove the optimality of MTS-B.

**THEOREM 1** (Optimality of MTS-B Algorithm). *With MTS-B algorithm, each node in a wireless topology  $G = (V, E)$  can*

compute the optimal forwarder lists and the corresponding number of transmissions to any other node.

*Proof:* Without loss of generality, we consider the proof for one destination  $d$ , which includes two steps. First, we prove that using the MTS-B algorithm, every node  $s$  can obtain sufficient conditions to compute the optimal forwarder list to  $d$ . Then, we prove that the number of transmissions for each node pair  $(s, d)$  decreasingly converge to the optimal solutions.

**(1) Obtaining sufficient conditions for optimality:** in this part, we prove by induction that using MTS-B algorithm, every node  $s$  can obtain sufficient conditions to compute the optimal forwarder list to  $d$ . Given a destination  $d$ , the optimal forwarder list of each possible source  $s \in V/\{d\}$  can be formulated as  $FL^s(d) = \{s, v_{m-1}, \dots, v_1, d\}$ , where  $1 \leq m \leq N - 1$ . The length (or the maximum hop-count) of forwarder list is  $m$ . For each source  $s$  with  $m = 1$ ,  $s$  can simply obtain the optimal forwarder list to  $d$  as  $\{s, d\}$  in initialization process of Algorithm 2 and the expected number of transmission as  $N^s(d) = 1/Pr(s, d)$ . Similarly, the source  $s$  with  $m = 2$ , has the optimal forwarder list in form of  $FL^s(d) = \{s, v_1, d\}$ . Based on Lemma 1,  $s$  only needs to get the optimal forwarder list information of  $v_1$  to compute its own optimal forwarder list. Moreover, also from Lemma 1, we know that  $v_1$  has to be the node with  $m = 1$  —which is already optimized in initialization process. Hence, the optimal forwarder list of nodes with  $m = 2$  can be computed.

Now, we consider the sources with  $m = k$ ,  $3 \leq k \leq N - 1$ , namely with optimal forwarder list  $FL^s(d) = \{s, v_{m-1}, \dots, v_1, d\}$ . We assume that every node with  $m < k$  has computed its optimal forwarder list to  $d$ . From Lemma 1, we know that every node  $v_i \in FL^s(d)$  has the optimal forwarder list with length less than  $k$ . Hence,  $s$  can compute its optimal forwarder list to  $d$  upon receiving the optimal forwarder list information from neighbors  $v_i$ 's, where  $v_i \in B_s \cap FL^s(d)$ .

**(2) Convergence of MTS-B algorithm:** when  $s$  receives an update message from its neighbor  $B_s(i)$ , the message gets processed if and only if the expected number of transmissions  $N^{B_s(i)}(d)$  carried in the message satisfies the condition

$$N^{B_s(i)}(d) < N^s(d). \quad (4)$$

Once condition eq. (4) is satisfied, the forwarder list  $FL^s(d)$  from the sender  $s$  to the destination  $d$  gets updated to  $\mathbf{merge}\{FL^{B_s(i)}(d), FL^s(d)\}$  or is replaced by  $\{s, FL^{B_s(i)}(d)\}$ , which is determined by which case leads to lower expected number of transmissions from  $s$  to  $d$ . Moreover, using eq. (3) and eq. (4), we have

$$\mathcal{N}(\mathbf{merge}\{FL^{B_s(i)}(d), FL^s(d)\}) < \mathcal{N}(FL^s(d)) = N^s(d). \quad (5)$$

Hence, if the event  $\mathcal{N}(\mathbf{merge}\{N^{B_s(i)}(d), FL^s(d)\}) > \mathcal{N}(\{s, FL^{B_s(i)}(d)\})$  happens, based on eq. (5), we have

$$\mathcal{N}(\{s, FL^{B_s(i)}(d)\}) < N^s(d), \quad (6)$$

which basically means that the expected number of transmissions from  $s$  to  $d$  is decreased after the update. On the other hand, if the event  $\mathcal{N}(\mathbf{merge}\{FL^{B_s(i)}(d), FL^s(d)\}) <$

$\mathcal{N}(\{s, FL^{B_s(i)}(d)\})$  happens, eq. (5) directly shows us the decrease of the expected number of transmission after the update. All in all, we proved that when  $s$  receives an update message from its neighbor, the updated forwarder list leads to decreased expected number of transmissions from  $s$  to  $d$ .

Therefore, combining the results in (1) and (2), we proved that the MTS-B algorithm can converge to the optimal forwarder list for each source-destination pair  $(s, d)$ . ■

## V. CLOSE-NODE-SET THEORY

In this section, we present the close-node-set (CNS) partitioning scheme for decomposing the wireless topology into several sub-topologies, so that node pairs within the same sub-topology can find the global optimal forwarder lists to each other using the local information. Then, we propose the localized opportunistic routing protocol that enables scalability.

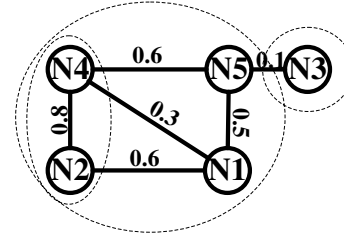


Fig. 2. Nested CNSs with link qualities marked on the links.

### A. CNSs Construction

*1) Definition of CNS:* Given a wireless topology  $G = (V, E)$ , the optimal transmission cost matrix can be computed using MTS or MTS-B, where each entry in that cost matrix indicates the estimated number of transmissions needed for sending a single packet from source (row index) to destination (column index). For a specific destination  $d$ , all of the possible sources could be ordered or ranked by their optimal expected numbers of transmissions to  $d$ , represented by the column with index  $d$ . Lower transmission cost results in higher ranking. Then, we define the ranking matrix. Note that here the rank represents the order of nodes in terms of their expected number of transmissions. It is *not* the matrix rank, i.e., the number of linearly independent rows or columns of the matrix. The rank  $R = (R_{ij})$  for connected wireless network, where each entry  $R_{ij}$  represents the order of node  $i$ 's in terms of the expected numbers of transmissions to destination node  $j$  among all nodes. For example, the optimal expected transmission cost matrix  $C$  and the corresponding ranking matrix  $R$  for the topology in Fig. 2 are

$$C = \begin{pmatrix} 0 & 1.6 & 12 & 2.0 & 1.9 \\ 1.7 & 0 & 13 & 1.3 & 2.8 \\ 12 & 12.6 & 0 & 12 & 10 \\ 2.1 & 1.3 & 12 & 0 & 1.7 \\ 2 & 2.6 & 10 & 1.7 & 0 \end{pmatrix}; R = \begin{pmatrix} 1 & 3 & 4 & 4 & 3 \\ 2 & 1 & 5 & 2 & 4 \\ 5 & 5 & 1 & 5 & 5 \\ 4 & 2 & 3 & 1 & 2 \\ 3 & 4 & 2 & 3 & 1 \end{pmatrix}.$$

In  $R$ , nodes are listed on the rows and columns with the same order  $\{N1, N2, N3, N4, N5\}$ . We could see that node  $N2$  and  $N4$  take each other as the highest rank node in their own (column) ranking vectors. This is, if any other node  $N_k$  tries to communicate with any node in  $\{N2, N4\}$ , it only needs to know how to reach this two nodes set — because any of  $\{N2, N4\}$  has higher rank than the source  $N_k$ . We could also see that the node set  $\{N2, N4, N1, N5\}$

has the same property. If we group this sort of nodes together and list the nodes on the rows and columns in order as  $\{N2, N4, N1, N5, N3\}$ , the ranking matrix becomes

$$R' = \begin{pmatrix} |\bar{1}| & |\bar{2}| & \bar{2} & |\bar{4}| & 5 \\ |\bar{2}| & |\bar{1}| & 4 & 2| & 3 \\ |\bar{3}| & 4 & 1 & 3| & 4 \\ |\bar{4}| & \bar{3} & \bar{3} & \bar{1}| & 2 \\ \bar{5} & \bar{5} & \bar{5} & \bar{5} & 1 \end{pmatrix}.$$

Now, we are in the position to give the definition of CNS.

**DEFINITION 1** (Close-node-set (CNS)). *Given a wireless topology  $G = (V, E)$ ,  $R$  is the ranking matrix. If for a node set  $\{v_1, v_2, \dots, v_n\}$  ( $n \leq |V|$ ),  $R(v_i, v_j) \leq n$  holds true for any  $i, j \leq n$ , the set  $\{v_1, v_2, \dots, v_n\}$  is a close-node-set.*

Each CNS obeys to the following two rules:

- **Rule 1:** If the source  $s$  and the destination  $d$  are in the same CNS, the forwarders of the optimal forwarder list are all located in the same set CNS.
- **Rule 2:** If the source  $s$  and the destination  $d$  are in the different CNSs (say  $CNS_i$  and  $CNS_j$ ), the source  $s$  only needs to know how to get to  $CNS_j$ , the destination belongs to. Once the packet reaches the destination set  $CNS_j$ , it can find the local optimal opportunistic paths to the destination based on Rule 1.

The necessary and sufficient conditions of CNS are given in Theorem 2 and Theorem 3.

**THEOREM 2** (Necessary condition of CNS). *Given a close-node-set  $CNS_k$ ,  $\forall v_i, d \in CNS_k$  and  $v_j \notin CNS_k$ , let  $N^d(v_i)$  and  $N^d(v_j)$  denote the optimal expected number of transmissions of  $v_i$  and  $v_j$  to  $d$ , respectively. We have*

$$N^d(v_i) < N^d(v_j). \quad (7)$$

*Proof:* Using definition of CNS, since any  $v_j \notin CNS_k$  has higher rank than any  $v_i \in CNS_k$ , the corresponding optimal expected number of transmissions  $N^d(v_i) < N^d(v_j)$  must hold. ■

**THEOREM 3** (Sufficient condition of  $k$  nodes CNS). *In a wireless topology  $G = (V, E)$ , let  $N_V^s(d)$  denote the global optimal expected number of transmissions from source  $s$  to destination  $d$ , with respect to the complete node set  $V$ . Given a node set  $C = \{v_1, \dots, v_k\}$  ( $k < |V|$ ), if for any  $v_i \in C$  and any  $C$ 's one-hop neighbor node  $u \in \bar{C}$ , we have*

$$N_{C \cup \{u\}}^u(v_i) \geq \max_{v_j \in C} N_C^{v_j}(v_i), \quad (8)$$

then the node set  $C$  is a close-node-set.

*Proof:* Let  $v_i, v_j \in C$  ( $i \neq j$ ) be any two different nodes in  $C$ , and  $u \in \bar{C}$  be any  $C$ 's one-hop neighbor node. If node  $v_i$  is the destination, based on Theorem 4 in [19], node  $v_j$  has higher rank in  $v_i$ 's ranking vector  $r(v_i)$  than node  $u$ , since we have  $N_{C \cup \{u\}}^u(v_i) \geq \max_{v_j \in C, j \neq i} N_C^{v_j}(v_i)$ .

If source node  $v_k$  is more than one hop away from  $C$ , it has to select at least one  $C$ 's one-hop neighbor node, say  $u \in \bar{C}$ , to reach the destination node  $v_i$ . In this case,  $v_k$ 's rank in ranking vector  $r(v_i)$  will be even lower than node  $u$ , based on the Theorem 4 [19]. Hence, we know that every node  $v \in C$  have lower rank than nodes  $v_j \in C$  in  $r(v_i)$ . We can also

get similar result when taking any other node  $v_j \in C$  as the destination. Therefore, node set  $C = \{v_1, \dots, v_k\}$  is a CNS, based on the CNS definition 1. ■

From Theorem 2 and Theorem 3, we conclude that any wireless topology can be viewed as several CNS's. For example, Fig. 3 shows a wireless topology with 4 nested CNS sets, each of which represents an equivalent node in the transferred overlay network (shown in Fig. 4). If the source  $s$  wants to send data to destination  $d$ , it just needs to find the CNS path from its own  $CNS_1^1$  to  $d$ 's  $CNS_2^1$ , as  $\{CNS_1^1, CNS_1^2, CNS_2^1\}$ . In section V-B, we will address the CNS path selection problem and propose the localized opportunistic routing (LOR) protocol. Now, we proceed to address the construction of CNSs.

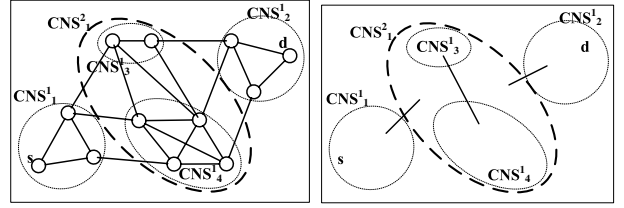


Fig. 3. CNS Graph Partitioning

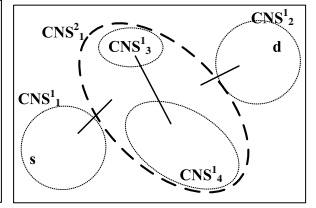


Fig. 4. Equivalent Overlay Network

2) *CNSs Construction:* Using MTS-B, each node in the wireless topology exchanges its own information with its neighbors. A node can construct a local CNS, once it collects sufficient local information which satisfies Theorem 3. Each CNS set creates a CNS id. This process ends when hop counts included in the local exchanged information exceed a previously assigned threshold  $S_{max}$ , which indicates the average number of neighbors in the wireless network. Note that  $S_{max}$  determines the maximal size of CNS constructed by the partitioning scheme. The expected CNS size is about one radio range large, so it depends on the average number of neighbors of the wireless network. Smaller  $S_{max}$  leads to less control overheads. The extreme case is  $S_{max} = 1$ , where there is no control overhead. Contrarily, if  $S_{max}$  is too large, the control overheads would approach the overhead as in the global optimal algorithms, like MORE, ExOR, and MTS, since every node needs to collect information of the entire topology.

Each node in the CNS maintains a local optimal forwarder lists information to any other member of the same CNS, and also maintains a CNS routing table, which lists the neighbor CNS id's and the corresponding best interface node. If there are multiple interface nodes to the same neighbor CNS, we will select the one, with the least number of transmissions to the neighbor CNS, as the best interface node. The expected number of transmissions from a node  $v_j$  to a neighboring CNS $_i$  can be computed as

$$N^{v_j}(CNS_i) = \frac{1}{1 - \prod_{v_k \in CNS_i \cap B_{v_j}} (1 - Pr(v_j, v_k))}, \quad (9)$$

which indicates the expected number of transmissions needed for  $v_j$  to transmit one packet to at least one node in CNS $_i$ . Notice that in our protocol, and since we use a fixed  $S_{max}$ , the relative size of each CNS is almost same, and that edge case of having one CNS noticeably larger than the other is an excluded edge case. Accordingly, choosing the shortest path

length by the number of of CNS's in it would imply a shortest overall path, since every CNS is almost equal in size.

**Naming of CNS's:** We assume that every wireless node has an unique identification (ID), which can be easily achieved, for example, by using the MAC address, or pre-assigned ids. During the CNS construction procedure, CNS's do not have global knowledge of the entire network. Hence, the challenge of CNS naming is how to assign CNS id's so no overlapped CNS ids exist. In our algorithm, once a CNS is formed, it chooses one of the wireless nodes in that CNS, and takes the node id as its CNS id. Moreover, since each CNS has full knowledge of all nodes and CNS's in it, a nested CNS, within which there is at least one other CNS inside, will take an id of its wireless nodes, that has not been taken by other CNS's in it. Hence, the CNS id generation is done during the CNS construction procedure, at no additional overhead.

### B. LOR: Localized Opportunistic Routing Protocol

Since any network could be divided into several nested autonomous CNS's, each CNS can be taken as an equivalent node, and the topology is transformed to an overlay with only CNS's in it. Then, we exploit the single path routing idea in the overlay network to find the single CNS path.

**Control Plane:** We adopt the routing request mechanism used in traditional single path wireless routing protocols, e.g., AODV[30], to find CNS path. When the source node  $s$  wishes to send packets to a certain destination  $d$ , it broadcasts a CNS route request message containing: the source id, the destination id, the life time of the message, a message ID, and a node sequence, where ids of subsequent nodes receiving this message will be appended to the end of the node sequence. a CNS sequence will also be recorded indicating what CNS path a message has taken. Once the destination receives the first copy of route request message, it creates and returns a route reply message, which includes the CNS sequence recorded by the request message. The reply message will be sent back through the path recorded by the node sequence in the request message. This mechanism has two nice features. First, a node  $v$  will not repeat a route request that has been received or created by  $v$ . Moreover, the life time of the message specified by the source limits how many times they can be retransmitted.

**Data Plane:** Once the source receives the CNS path, it locally sends packets to the next-hop CNS with opportunistic routing. First, it checks the local CNS routing table and finds the corresponding best interface node to the next-hop CNS set. Then it takes the interface node as the local destination and constructs the local forwarder list. The current CNS set might have several interface nodes to the next-hop CNS. During the local opportunistic routing, the lower priority node will not re-forward the packet which has reached a higher priority nodes or the next-hop CNS. In this way, we can guarantee that the best interface node has the highest chance to forward the packet to the next hop CNS, whereas we also utilize all opportunities of other non-best interface forwarders to help delivering the packet to the next-hop CNS. Once the packet reaches an intermediate CNS, the receiver checks the next hop CNS information carried on the packet, and similarly finds the best interface node to construct the local opportunistic forwarder list for delivering the packet.

### C. Discussion

Recall that the distributed MTS-B algorithm has time complexity  $\mathcal{O}(N^2)$  with  $N$  as the total number of nodes. In section V, we design a clustering method that decompose the large scale wireless network into small pieces, within which node pairs have equal global and local optimal forwarder lists. This way, the distributed MTS-B algorithm only needs to explore and construct local CNS's instead the entire network, which significantly reduce the time complexity of forwarder list selection algorithm and make it more scalable in terms of the network size. Hence, our algorithm is applicable in many high mobility large-scale wireless network environments, such as multi-hop mobile ad hoc networks and vehicular networks. Taking multi-hop vehicular networks (VANETs)[30], [35], [18] as an example, it is designed to efficiently disseminate warning information to incoming vehicles, about road congestion, construction, and safety warning, etc. Our localized opportunistic routing protocol can dynamically group local vehicles into small CNS sets with low cost, i.e.,  $\mathcal{O}(N_{loc}^2)$  time complexity, where  $N_{loc} \ll N$  is the size of the local CNS. The low complexity guarantees the fast convergence and makes it adaptive to high mobility wireless scenarios.

To highlight the advantages of our local opportunistic routing (LOR) protocol, Table III compares LOR with MTS and MTS-B schemes in terms of the time complexity (or scalability), optimality, and algorithm design pattern.

TABLE III  
COMPARING LOR, MTS, AND MTS-B ALGORITHMS

Schemes	MTS	MTS-B	LOR
Complexity	$\mathcal{O}(N^2)$	$\mathcal{O}(N^2)$	$\mathcal{O}(N_{loc}^2), N_{loc} \ll N$
Optimality	Global optimal	Global optimal	Local optimal
Design	Centralized	Distributed	Distributed

## VI. PERFORMANCE EVALUATION

In this section, we present evaluation results of LOR on ad hoc wireless networks and vehicular networks, respectively.

### A. Evaluations on ad hoc wireless networks

We conduct simulations in ns-2 [10] to evaluate the performance of LOR and compare it with ExOR and MORE. The simulation parameters are in Table IV. UDP is used for realtime traffic. We use realtime CBR (constant bit rate) flow with 512 Byte packets. Below, we will evaluate its performances in control plane and data plane, respectively.

**Control plane.** Both MTS and ETX require whole topology information, thus produce the same control messages at each node. Now, we compare the control overheads of MTS/ETX schemes with our LOR protocol.

TABLE IV  
SIMULATION PARAMETERS

Parameter	Values
Bandwidth	1Mbps
MAC layer	802.11b
Routing protocol	LOR, ExOR and MORE with MTS,
Transmission protocol	UDP

1) *How does network diameter affect the control overheads?* In the first set of results, we test how the network density affects the transmission control overheads. The density is measured by the average number of neighbors for each node in the

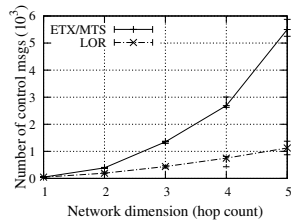


Fig. 5. Control overhead vs network diameter.

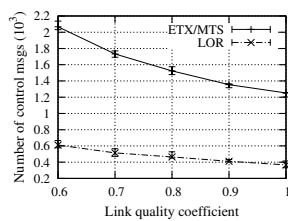


Fig. 6. Control overhead vs link quality.

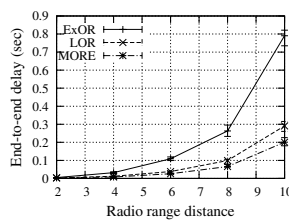


Fig. 7. End-to-end delay vs radio range distance.

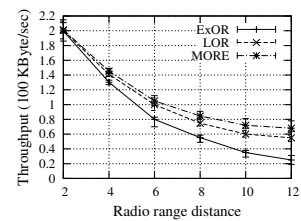


Fig. 8. Throughput vs radio range distance.

wireless network. We had the wireless network density as 5 neighbors per node, thus the maximal CNS size  $S_{max} = 5$ . We then changed the network diameter (defined as the greatest shortest path distance (in hop counts) between any pair of nodes) from 1 to 5 hops, to evaluate the control overheads. For each diameter, we randomly generated 30 topologies, and performed 100 simulations with different random seeds for each topology. To do that, we generated topology with 60 nodes, and randomly placed wireless links with constraint of the pre-determined network diameter (1 . . . 5). The link quality of each present link is uniformly generated from the range of  $[0, 1]$  at random. Fig. 5 shows that as the diameter increases, the control messages increase as well. This happens because when the diameter increases, the number of nodes in the topology also increases in order to maintain the same density. Furthermore, with LOR, a node only needs to collect local area information with at most  $S_{max} = 5$  nodes, whereas ETX/MTS needs to collect all information of the entire topology. As shown in Fig. 5, the number of control overheads of MTS/ETX grows much faster than LOR, and it is about 5 times the overheads of LOR when the diameter is 5 hops.

2) *How do link qualities affect the control overheads?* In the second set of simulations, we randomly generated one wireless topology, with 100 nodes. The diameter is 3 hops, and its density is 5 neighbors per node. We use the same scale factor to all of the link qualities, and vary the scale factor as 0.6, 0.7, 0.8, 0.9, and 1.0 — to test how the link qualities affect the control overheads. Fig. 6 shows that as the link qualities increase, the control overheads decrease. This is because higher link qualities result in less number of retransmissions of control messages. We also see that LOR always has lower control overheads than MTS/ETX — which is because nodes that use LOR only need sufficient local information to compute the forwarder lists.

**Data plane.** In the data plane comparison, we mainly consider two performance metrics, the end-to-end delay and the throughput, where the throughput is measured by the number of bytes transmitted from the source to the destination per unit time (in KB/sec). We implement the centralized MTS algorithm in ExOR and MORE respectively. We compare each of these two schemes with LOR. In the simulations, we randomly generated 30 wireless topologies with different sizes from 60 to 90 nodes, diameters from 1 to 5. Again, the quality of each link is uniformly chosen from  $[0, 1]$  at random. Then, we randomly chose source-destination pair from those topologies to test the communication performances. We run 100 simulations with different random seeds for each

node pair. Fig. 7 shows the end-to-end delay performance comparison. Assuming that all nodes in the topology have the same radio range, we define radio range distance between two wireless nodes as the distance in terms of number of radio ranges, namely, the physical distance between two nodes divided by the radio range. For example, when a single radio range is 100 meters, two nodes that are 300 meters away are considered as 3 radio ranges distant from each other. Hence, radio range distance is a distance measure of a node pair, which captures the actual distance between node, and does not reflect the network density, because two node in 2 radio ranges may have 20 nodes between them serving as relays, or have no relay node in-between at all. On the other hand, we used network diameter in Control plane evaluation—a network metric, measuring the average hop counts to reach one node from the other. We could see that as the radio range distance between the source and destination increases, the end-to-end delay also increases. MTS-based MORE performs best. LOR has a little higher end-to-end delay than MORE, and ExOR has the worst performances among them. This happens because MORE and LOR can realize spatial reuse, and MORE can realize global optimal forwarder list. However, comparing with the control overheads reduced by LOR, this end-to-end delay cost is acceptable. Similar results are obtained when comparing the throughput (see Fig. 8). MTS-based MORE performs slightly better than LOR and the performances of MTS-based ExOR is the worst among them.

### B. Evaluations in vehicular networks

As an practical implementation of mobile ad hoc networks, vehicular ad hoc networks (VANETs) are generating great interest in research community because of their promise for improving driving experience with respect to both safety and convenience concerns. In VANETs, each moving vehicle acts as a wireless node in the wireless network and it can communicate with vehicles within the radio range. Different from other types of wireless networks, in VANET environment the vehicle mobility patterns follow a geographical topology and vehicle movement is restricted by road map as well as traffic regulations (e.g., the traffic lights and speed limits).

Routing is an important operation in VANETs and is used, for example, for exchanging GPS and notification information among vehicles to select better driving paths. The carry-and-forward techniques are proposed in [37], [29], where vehicles carry or forward packets progressively close to an access point by selecting potential shortest path. More recent works utilize additional trajectory information to further improve the routing performances [13], [14], [34], [33].



In this section, we evaluate how our local opportunistic routing protocol performs in VANETs, by comparing it to the Vehicle-assisted data delivery (VADD) algorithm [37] under the condition that no additional trajectory information are available. Note that when additional information is available, it is interesting to study how to incorporate those information with our LOR algorithm to further improve the routing efficiency. Since this paper focuses on mainly applications in general wireless networks, we leave this aspect to our future work, and conduct this simulation under fair comparison settings where our algorithm is applicable. Below, we first briefly introduce VADD algorithm, and then detail our simulation settings and finally present the results.

**Vehicle-assisted data delivery (VADD) protocol.** [37] proposes the VADD protocol which uses a stochastic model with vehicular traffic statistics for data forwarding. Given a packet carrier and a destination, e.g., a vehicle, traffic lights, or traffic cameras, VADD performs in two phases (at intersections and on straight ways, respectively) iteratively to deliver the packet to the destination. By taking the underlying road map as a graph, with intersections as nodes and roads as edges, at each intersection VADD utilizes the GPS map and traffic information to calculate the best next hop, i.e., the next intersection, to send the packet to and to decide which carrier to forward the packet to so as to reduce the delay to the next intersection. On the other hand, when the carrier is on a straight way, a greedy perimeter stateless routing (GPSR) [17] is used to deliver the packet toward the predetermined next intersection, and the current packet carrier continues to carry the packet, if there is no vehicle to forward ahead. While applying our LOR algorithm in VANETs, the GPS map and traffic information is utilized to calculate the Close-node-set path, instead of the traditional single path wireless routing.

**Simulation results.** We use SUMO [1] for road map generation, which provides three different ways for generating the road map by the users requirements, automatic generation or import of existing real world maps. It also allows users to specify the vehicles' flow, trip and route. To simulate a large scale scenario, we generate a grid road network with 10 horizontal streets by 15 vertical streets (on a  $10km \times 12km$  area), thus consisting of 150 intersections, where the rest of the simulation parameters are shown in Table V. Below we will investigate how vehicle velocity and number of vehicles affect the routing performance, in terms of the end-to-end delay.

TABLE V  
VANET SIMULATION PARAMETERS

Parameter	Values
Simulation Time	3600 seconds
# of vehicles	400–550
Simulation Area	$10km \times 12km$
Number of Intersections	150
Radio range	250m
Bandwidth	20Mbps
Packet size	1000 Bytes
Velocity	10 to 60 miles/hr

Figures 9 and 10 show how the end-to-end (E2E) delay and network throughput change when varying the average vehicle velocity, with 460 vehicles. We observe that as the average vehicle velocity increases, the end-to-end delay of LOR increases

and the throughput of LOR decreases, which is anticipated, and is in part due to the more frequent recalculations of the close-node-set. In contrast, as the average vehicle velocity increases, the end-to-end delay of VADD protocol decreases, and the network throughput increases, which is because of the high vehicle speed that yields high vehicle arrival rate at each road segment, and leads to the shorter delivery delay and higher throughput. Overall, we observe that LOR still performs better than VADD even when the average velocity of the vehicle is as high as 60 miles per hour.

Figures 11 and 12 present how the E2E delay and throughput change with the number of vehicles (NoV) ranging from 400 to 550 and the average velocity fixed as 30 MPH. It is clear that as the average number of vehicles increases, the end-to-end delays of both LOR and VADD decrease, while both throughputs increase, which is because more vehicles preserve more stable close-node-set in LOR, and bring more opportunities for vehicles to encounter in VADD. However, the opportunistic mechanism in LOR still outperforms the carry-and-forward scheme exploited in VADD. For example, even when both schemes are operated for 550 vehicles, the average end-to-end delay in LOR is only 50% of that in VADD, while the throughput of LOR is around thrice of that in VADD.

## VII. CONCLUSION

The opportunistic routing protocols require whole topology information to construct global optimal forwarder list, which restricts their applicability in large-scale wireless networks. In this paper, we first introduce the distributed minimum transmissions selection (MTS-B) algorithm which can produce the optimal forwarder list for each source-destination pair in a distributed manner. Then, by utilizing the characteristics of opportunistic routing, we establish the close-node-set (CNS) theory that partitions wireless topologies into several nested node sets with local information — which can realize optimal opportunistic routing in each local set. Moreover, using these results, we propose the localized opportunistic routing (LOR) protocol suitable for large-scale wireless networks. Extensive simulation results demonstrate that LOR can dramatically reduce the control overheads over current opportunistic routing protocols as the wireless network scales up.

As part of our future work, we are planning to further study the large-scale opportunistic routing by explicitly taking the asymmetric wireless channels into consideration and explore how to improve the routing performance by exploiting the theories on directed graphs [22], [21], [24], [23]. Moreover, we are also interested in considering both end-to-end delay and wireless node energy consumption in designing high performance large-scale opportunistic routing (e.g., [25], [28]).

## REFERENCES

- [1] Simulation of Urban MObility (SUMO). <http://sumo.sourceforge.net/>.
- [2] U. G. Acer, S. Kalyanaraman, and A. A. Abouzeid. Weak state routing for large scale dynamic networks. In *ACM MOBICOM*, 2007.
- [3] S. Biswas and R. Morris. Exor: Opportunistic routing in multi-hop wireless networks. In *ACM SIGCOMM*, 2005.
- [4] S. Chachulski, M. Jennings, S. Katti, and D. Katabi. Trading structure for randomness in wireless opportunistic routing. In *ACM SIGCOMM*, pages 169–180, 2007.
- [5] D. D. Couto, D. Aguayo, J. Bicket, and R. Morris. High-throughput path metric for multi-hop wireless routing. In *MOBICOM*, Sep 2003.

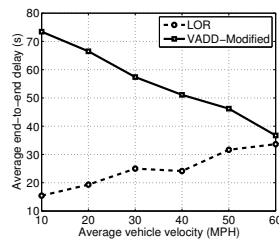


Fig. 9. E2E delay vs velocity

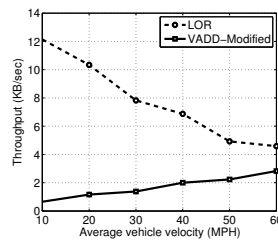


Fig. 10. Throughput vs velocity

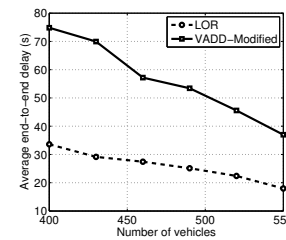


Fig. 11. E2E delay vs NoV

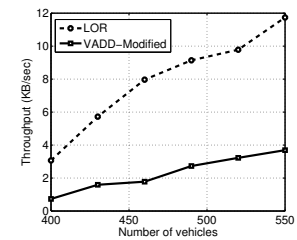
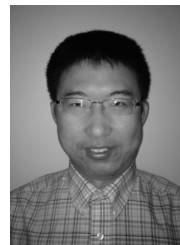


Fig. 12. Throughput vs NoV

- [6] E. M. Daly and M. Haahr. Social network analysis for routing in disconnected delay-tolerant manets. In *ACM MobiHoc*, 2007.
- [7] H. Dubois-Ferriere, M. Grossglauser, and M. Vetterli. Age matters: efficient route discovery in mobile ad hoc networks using encounter ages. In *ACM MobiHoc*, pages 257–266, 2003.
- [8] H. Dubois-Ferriere, M. Grossglauser, and M. Vetterli. Least-cost opportunistic routing. In *Allerton Conference*, 2007.
- [9] V. Erramilli, A. Chaintreau, M. Crovella, and C. Diot. Diversity of forwarding paths in pocket switched networks. In *ACM IMC*, 2007.
- [10] K. Fall and K. Varadhan. *The ns-2 Manual*. The VINT Project, UC Berkeley, LBL, and Xerox PARC, 2003.
- [11] M. Grossglauser and M. Vetterli. Locating mobile nodes with ease: Learning efficient routes from encounter histories alone. *IEEE/ACM Transaction on Networking*, 14(3):457–469, 2006.
- [12] C. Hedrick. RFC 1058: routing information protocol, 1988.
- [13] J. Jeong, S. Guo, Y. Gu, T. He, and D. Du. Tbd: Trajectory-based data forwarding for light-traffic vehicular networks. In *ICDCS*, pages 231–238. IEEE, 2009.
- [14] J. Jeong, S. Guo, Y. Gu, T. He, and D. Du. Tsf: Trajectory-based statistical forwarding for infrastructure-to-vehicle data delivery in vehicular networks. In *IEEE ICDCS*, pages 557–566, 2010.
- [15] D. B. Johnson and D. A. Maltz. Dynamic source routing in ad hoc wireless networks. In *Mobile Computing*. Kluwer Publishers, 1996.
- [16] E. P. C. Jones, L. Li, and P. A. S. Ward. Practical routing in delay-tolerant networks. In *Workshop on Delay-tolerant networking*, 2005.
- [17] B. Karp and H. Kung. Gpsr: Greedy perimeter stateless routing for wireless networks. In *ACM MOBICOM*, pages 243–254, 2000.
- [18] F. Li and Y. Wang. Routing in vehicular ad hoc networks: A survey. *Vehicular Technology Magazine, IEEE*, 2(2):12–22, 2007.
- [19] Y. Li, W. Chen, and Z.-L. Zhang. Optimal forwarder list selection in opportunistic routing. In *MeshTech*, pages 670–675. IEEE, 2009.
- [20] Y. Li, Y.-A. Liu, L. Li, and P. Luo. Local scheduling scheme for opportunistic routing. In *IEEE WCNC*, 2009.
- [21] Y. Li and Z.-L. Zhang. Random walks on digraphs: A theoretical framework for estimating transmission costs in wireless routing. In *IEEE INFOCOM*, pages 1–9, 2010.
- [22] Y. Li and Z.-L. Zhang. Random walks on digraphs, the generalized digraph laplacian and the degree of asymmetry. In *WAW*, 2010.
- [23] Y. Li and Z.-L. Zhang. Digraph laplacian and the degree of asymmetry. *Internet Mathematics*, 8(4), 2012.
- [24] Y. Li and Z.-L. Zhang. Random walks and green’s function on digraphs: A framework for estimating wireless transmission costs. *IEEE/ACM Transactions on Networking*, PP(99):1–14, 2012.
- [25] Y. Li, Z.-L. Zhang, and D. Boley. The routing continuum from shortest-path to all-path: A unifying theory. In *IEEE ICDCS*, 2011.
- [26] Y. Li, D. Zou, H. Wang, Z. Zhou, Y.-A. Liu, and Y. Qiao. LOR: Localized opportunistic routing in large-scale wireless network. In *Globecom*, pages 1–5. IEEE, 2010.
- [27] Y. Lin, B. Li, and B. Liang. Codeor: Opportunistic routing in wireless mesh networks with segmented network coding. In *ICNP*, 2008.
- [28] H. Liu, Z.-L. Zhang, J. Srivastava, and V. Firoiu. Pwave: A multi-source multi-sink anycast routing framework for wireless sensor networks. In *Networking*, pages 179–190. Springer, 2007.
- [29] L. Pelusi, A. Passarella, and M. Conti. Opportunistic networking: data forwarding in disconnected mobile ad hoc networks. *Communications Magazine, IEEE*, 44(11):134–141, 2006.
- [30] C. E. Perkins and E. M. Royer. Ad-hoc on-demand distance vector routing. In *Mobile Computing Systems and Applications*, 1999.
- [31] T. Spyropoulos, K. Psounis, and C. S. Raghavendra. Efficient routing in intermittently connected mobile networks: the single-copy case. *IEEE/ACM Trans. Netw.*, 16(1):63–76, 2008.
- [32] R. Wattenhofer, L. Li, P. Bahl, and Y. Wang. Distributed topology control for power efficient operation in multihop wireless ad hoc networks. In *IEEE INFOCOM*, 2001.
- [33] Y. Wu, Y. Zhu, and B. Li. Infrastructure-assisted routing in vehicular networks. In *IEEE INFOCOM*, 2012.
- [34] F. Xu, S. Guo, J. Jeong, Y. Gu, Q. Cao, M. Liu, and T. He. Utilizing shared vehicle trajectories for data forwarding in vehicular networks. In *IEEE INFOCOM*, pages 441–445, 2011.
- [35] J. Yu and P. Chong. A survey of clustering schemes for mobile ad hoc networks. *IEEE Comm. Surveys & Tutorials*, 2005.
- [36] K. Zeng, W. Lou, and H. Zhai. Capacity of opportunistic routing in multi-rate and multi-hop wireless networks. *IEEE Transactions on Wireless Communications*, 7(12-1):5118–5128, 2008.
- [37] J. Zhao and G. Cao. Vadd: Vehicle-assisted data delivery in vehicular ad hoc networks. *IEEE Transactions on Vehicular Technology (TVT)*, 57(3):1910–1922, 2008.
- [38] Z. Zhong and S. Nelakuditi. On the efficacy of opportunistic routing. In *IEEE SECON*, 2007.



**Yanhua Li** (S’09) received the B.S. and M.S. degrees in electrical engineering from Sichuan University, Chengdu, China, in 2003 and 2006, respectively, and the Ph.D. degree in electrical engineering from Beijing University of Posts and Telecommunications, Beijing, China, in 2009. He is currently pursuing a Ph.D. degree in computer science at University of Minnesota, Twin Cities.

His research interests are wireless networking and social network analysis, including communication protocols design and high performance networking scheduling strategies, social graph analysis and measurements.



**Abdelaziz Mohaisen** (S’09-M’12) obtained his M.S. and Ph.D. degrees in Computer Science from the University of Minnesota, both in 2012.

In 2012, he joined Verisign Labs where he is current a Research Scientist. Before pursuing graduate studies at Minnesota, he was a Member of Engineering Staff at the Electronics and Telecommunication Research Institute, a large research and development institute in South Korea. His research interests are in the areas of networked systems, systems security, data privacy, and measurements.



**Zhi-Li Zhang** received the B.S. degree in computer science from Nanjing University, Jiangsu, China, in 1986, and the M.S. and Ph.D. degrees in computer science from the University of Massachusetts Amherst, Amherst, in 1992 and 1997, respectively.

In 1997, he joined the Computer Science and Engineering faculty at the University of Minnesota, Minneapolis, MN, where he is currently a Professor. From 1987 to 1990, he conducted research at the Computer Science Department, Aarhus University, Aarhus, Denmark, under a fellowship from the Chinese National Committee for Education. He held visiting positions with Sprint Advanced Technology Labs, Burlingame, CA; IBM T. J. Watson Research Center, Yorktown Heights, NY; Fujitsu Labs of America, Sunnyvale, CA; Microsoft Research China, Beijing, CN; and INRIA, Sophia-Antipolis, France.