Utility-Based Opportunistic Routing in Multi-hop Wireless Networks

Jie Wu, Mingming Lu, and Feng Li
Department of Computer Science and Engineering
Florida Atlantic University
Boca Raton, FL 33431
{jie@cse., mlu2@, fli4@}fau.edu

Abstract—Recently, opportunistic routing (OR) has been widely used to compensate for the low packet delivery ratio of multi-hop wireless networks. Previous works either provide heuristic solutions without optimality analysis, or assume that unlimited retransmission is available for delivering a data packet. In this paper, we apply OR to a utility-based routing where the successful delivery of a data packet generates benefit. The objective is to maximize utility, defined as a function of benefit and cost of transmission. As the link reliability of each relay determines eventual packet delivery and hence utility, OR offers the ability to increase reliability through opportunistic relays. We explore the optimality of utility-based routing through OR without allowing retransmission, and observe that the optimal scheme requires exhaustive searching of all paths from source to destination. We then propose a heuristic solution to select relays and determine priorities among them. Finally, we provide distributed implementations for both schemes. Simulations on NS-2 and our customized simulator are conducted to verify the effectiveness of the heuristic compared with the optimal.

I. Introduction

Over the past decade, much research has been conducted on multi-hop wireless networks, which include wireless mesh networks [3], due to their broad applications and the easy deployment at low cost without relying on the existing infrastructure. However, this infrastructure-less property and the unstable nature of the wireless medium incur the problem of unreliable communication. To address this problem, OR [4] has been proposed as a new routing paradigm. OR utilizes the broadcast advantage of wireless communication to increase the reliability of a single transmission. Instead of relying on one next-hop node to forward a data packet (or simply, packet), OR pre-determines a set of candidate relays (or simply, relays) with a priority order and selects the highest-priority relay that indeed receives the packet as the actual forwarder, based on the instantaneous channel conditions and node availability.

Most existing OR protocols [4], [6] propose relay selection and prioritization based on some heuristics, and in general fail to provide optimality on the end-to-end performance. A recent work [5] proposes an OR scheme that can minimize the expected number of transmissions (ENT) to guarantee delivery. This optimization result is based on a strong assumption: unlimited retransmissions. Also, it is always better for a node

to select the neighbors with smaller ENTs as relays, and prioritizes relays based on their ENTs.

In this paper, we consider a new optimization scheme under the OR framework for a special utility-based routing scheme [10], where the successful delivery of a packet generates benefit (represented by a benefit value). The objective is to maximize utility, defined as a function of benefit and cost of transmission. More specifically, if the packet is successfully delivered, the corresponding utility is the benefit minus the cost of transmission. As wireless transmission is unreliable, packet delivery may fail, generating zero benefit. The benefit can be termed as expected benefit, and the same applies to utility. The cost is also termed as expected cost, as it depends on the actual occurrence of each transmission (although it does not depend on eventual delivery). Clearly, link reliability plays an important role in utility. High link reliability generates high utility under the same conditions of network topology and benefit value. It is interesting to note that in optimal utility-based routing, the optimal route depends not only on network topology (including reliability distribution), but also on the benefit value of the packet. As the link reliability of each relay determines eventual packet delivery and hence utility, OR offers to increase reliability through opportunistic relays.

The objective of this work is to maximize the utility rather than minimizing the expected cost to guarantee delivery. The challenge of our objective lies in the observation that relay selection is not confined to neighbors that are "closer" to the destination as in most current OR research. In addition, the optimal scheme requires exhaustive searching of all paths from source to destination. The computational complexity to find the optimal solution is very high. To reduce the complexity, we design a heuristic solution that uses the expected utility from a relay to the destination as a hint to select relays and determine priorities among them. We also implement both our optimal and heuristic solutions in a distributed way.

The main contributions of this paper can be summarized as follows: 1) We explore the optimality of OR without retransmission, propose a utility-based OR model, and prove that the optimal solution has to exhaustively search all paths from source to destination. 2) We propose both optimal and heuristic solutions, and implement them in a distributed way. 3) We verify the effectiveness of the heuristic solution through simulations on NS-2 and our customized simulator.

The following assumptions are used in this paper: 1) Each

Fig. 1. An example of opportunistic routing.

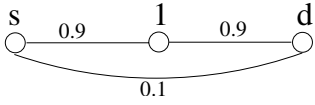|	u_d	u_1	u_2	U
r_1	20/30	15/24		10*/17.2
r_2	20/30		15/24	9.5/17.6*
r_3	20/30	8.5/14.8	15/24	4.8/9.8
r_4	20/30	15/24	8.5/14.8	3.7/9.3

Fig. 2. An example of a simple network.

node has a priori knowledge about its associated link cost and link reliability. Numerous works [15], [16] address the issue on how to collect this information. 2) In our model, the transmission range is adjustable. 3) Retransmission is not considered since retransmission may reduce utility in the absence of guaranteed eventual delivery.

The remainder of this paper is organized as follows. Some preliminaries are presented in Section II. Section III introduces the utility-based OR model, and proposes both the optimal solution and heuristic solution. Section IV empirically evaluates performance. Section V discusses related work. Finally, Section VI concludes this work and outlines future work. Due to space limitations, we omit all the proofs of the theorems in this paper.

II. PRELIMINARY

A. Opportunistic Routing Schemes

The underlying idea behind OR is that a stable route is likely available in unreliable multi-hop wireless networks if each node can utilize its multiple neighbors as potential forwarders. For example, in Figure 1, although the link between s and d is highly unreliable, node 1 is likely to receive the packet sent by s, and forward it to d. To find a stable and least-cost (in terms of expected number of transmissions, ENT) route, ExOR [4] specifies two rules: the rule of the selection of a *relay set* for each node, and the rule of *relay prioritization*. More specifically, ExOR can be summarized as follows: 1) A node not only selects a relay set but also assigns a priority to each relay according to ENT. 2) The node includes the IDs of the relays into the packet header in the order of priority and forwards the packet. 3) Upon receiving the packet, a relay schedules the timing to send an acknowledgement. The scheduled timing is reversely proportional to the priority. 4) Upon timeout, a relay decides to forward the packet if and only if no higher-priority node has sent an acknowledgement.

ExOR cannot guarantee that no redundant copies of the same packet will be transmitted in the network because not all relays of a node can receive the acknowledgement from the relay with higher priority. Therefore, we set the transmission range of the acknowledgements to be twice that of the transmission range of the packets so that all relays can receive acknowledgements.

B. Utility Model and Metrics

In utility-based routing [10], a source s intends to send packets to a destination d through relays in a multi-hop wireless network. Each packet is assigned a benefit value, v. The transmission cost and reliability from s to d are denoted as c and p, respectively. If a transmission is successful, s will obtain benefit v, incur cost c, and its utility is $v - c$. Otherwise,
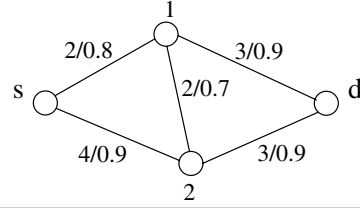
its utility is $0 - c$. Since the successful probability is p, and the failure probability is $1 - p$, the expected utility is:

$$U = p \cdot (v - c) + (1 - p) \cdot (0 - c) = p \cdot v - c. \quad (1)$$

In general, consider a multi-hop path $R, < s = 0, 1, 2, \cdots, m - 1, d = m >$. The corresponding expected utility is as follows:

$$U = (\prod_{j=0}^{m-1} p_{j,j+1}) \cdot v - \sum_{i=0}^{m-1} c_{i,i+1} \prod_{j=0}^{i-1} p_{j,j+1} = P_R \cdot v - C_R \quad (2)$$

where P_R is path reliability and C_R is path cost. Equation (2) can be derived through Equation (1) in a backward-fashion. For example, in Figure 2, four paths exist: $r_1 :< s, 1, d >$, $r_2 :< s, 2, d >$, $r_3 :< s, 1, 2, d >$, and $r_4 :< s, 2, 1, d >$. Each link is labeled with its associated cost/reliability. Considering path r_1, by applying Equation (2), we have $U = 0.8 \cdot 0.9 \cdot 20 - 2 - 3 \cdot 0.8 = 10$. We can also view node 1 as the virtual source and apply Equation (1) to link $(1, d)$: $u_1 = 0.9 \cdot 20 - 3 = 15$. We use u_i to represent the *residual expected network utilities* (RENU) of node i because node i is not the real source. Then, we can apply Equation (1) to link $(s, 1)$ by viewing 1 as the virtual destination: $U = 0.8 \cdot 15 - 2 = 10$.

The RENUs of the nodes on each path are listed in the table of Figure 2, where in each cell, two values separated by "/" represent the RENUs under benefit values of 20 and 30, respectively. If the benefit $v = 20$, the optimal path is $< s, 1, d >$, but if $v = 30$, the optimal route is $< s, 2, d >$. This reflects the property of utility-based routing: the optimal route depends not only on the network topology, but also on the benefit value. Based on the above observation, an optimal centralized algorithm and an approximation distributed algorithm were proposed to solve the routing problem [10].

III. THE PROPOSED MODEL AND SOLUTIONS

A. The Model

The metric for opportunistic routing. The *RENU under opportunistic routing* (OpRENU) scheme is different from the RENU scheme for a single multi-hop route. The calculation of OpRENU is based on all nodes in the relay set rather than a single downstream node in the calculation of RENU. We use opu_i to denote node i's OpRENU. Consider a node i with cost c (determined by its transmission range) and corresponding relay set. The relays are prioritized in order from $i+1$ to $i+k$
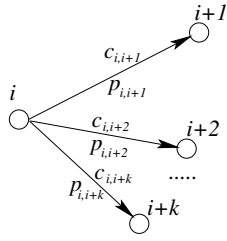
Fig. 3. An example of a node in the new model.

Algorithm 1 OpUtility(i)

1: Construct cost levels based on downstream neighbors;
2: **for** each cost level c **do**
3:     Insert downstream neighbor j (s.t. $c_{i,j} \leq c$) to relay set;
4:     Sort relays in the decreasing order of opu;
5:     Compute corresponding opu_i based on Formula (3);
6: Return maximal opu_i, corresponding cost level, relay set, and relay priorities;

---

with $i + 1$ as the highest-priority relay, as shown in Figure 3. With probability $p_{i,j}$, node j ($i + 1 \leq j \leq i + k$) receives a packet sent by node i. According to the OR framework described in Section II-A, node j forwards the packet if and only if all nodes with higher priorities fail to receive the packet and node j receives the packet. The corresponding probability is $p_{i,j} \cdot \prod_{l=i+1}^{j-1} (1 - p_{i,l})$. If the packet is forwarded by node j, the OpRENU contributed by node j should be $opu_j \cdot p_{i,j} \cdot \prod_{l=i+1}^{j-1} (1 - p_{i,l})$. Hence, node i's OpRENU can be calculated based on the OpRENUs of nodes in its relay set:

$$opu_i = \sum_{j=i+1}^{i+k} \left(opu_j \cdot p_{i,j} \cdot \prod_{l=i+1}^{j-1} (1 - p_{i,l}) \right) - c \quad (3)$$

We use the example in Figure 2 to illustrate Formula (3). The OpRENUs of the nodes are shown in the bottom table in Table I, where we assume that node 2 is node 1's relay. Comparing this table to the table in Figure 2, we can see that the utility of transmitting a packet with benefit value 20 (30) is increased from 10 (17.6) to 11.6 (20.9) in the sample network.

The relay set. As mentioned in Section II-A, two rules exist in OR: the rule for relay selection and the rule for relay prioritization. For any node i, the selection of relays should guarantee that no loop exists along any potential path through i. We refer to the neighbors that do not cause loops as *downstream neighbors*. Also, relay selection should consider transmission range, which determines the relay set from the downstream neighbors. Hence, in the relay selection of a node, we consider different transmission cost levels, which are directly proportional to the node's transmission range. The best transmission cost level should be the cost level that maximizes the OpRENU according to Formula (3).

For a given transmission cost level, relays are selected from the corresponding downstream neighbors, and the prioritization of the relays is based on the OpRENUs of the relays. The optimality of this prioritization rule can be guaranteed through the following result:
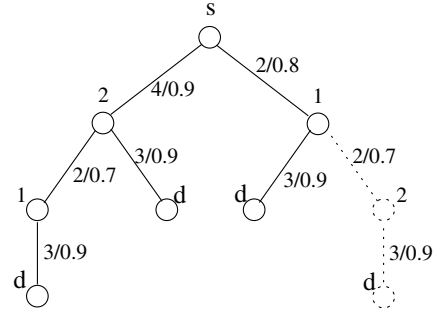

Fig. 4. The solution tree for the example in Figure 2.

Theorem 1. *(**Priority rule for relays**) For any node, once its relays have been determined, the relays should be prioritized in the decreasing order of their corresponding OpRENUs in order to maximize the OpRENU of the node.*

Relay selection and prioritization can be formalized through Algorithm 1. First, node i determines the possible cost levels from node i to its downstream neighbors. For each cost level c, node i constructs the corresponding relay set. A downstream neighbor j is included in the relay set if and only if $c_{i,j} \leq c$, where $c_{i,j}$ is the minimal power level to connect nodes i and j. After the relays have been sorted in the decreasing order of their corresponding OpRENUs, the OpRENU of node i can be calculated based on Formula (3). The best cost level that maximizes the OpRENU of node i, the corresponding relay set, the relay priorities, and the maximal OpRENU will be returned.

Algorithm 1 uses the downstream neighbors of a node and their corresponding OpRENUs as inputs. However, these inputs are not easy to obtain in the optimal solution shown in Section 3.2, since this requires the exhaustive exploration of all loop-free paths from source to destination. In the heuristic solution shown in Section 3.3, the downstream neighbors of node i are simply those neighbors that have larger RENUs than that of node i, and the OpRENUs of those downstream neighbors have been determined when node i is explored.

B. The Optimal Solution

As mentioned in Section I, the optimal solution for the OR problem should include all loop-free paths from source to destination, which is guaranteed by the following result:

Theorem 2. *(**Necessity of exhaustive searching**) In our OR model, the optimal solution should include all loop-free paths from source to destination.*

All the loop-free paths from source to destination can be described in an *optimal-solution tree* rooted at the source, where all leaf nodes are the destination. For a node i in the tree, all its neighbors, excluding those already appearing in the path from s to i, are listed as its children nodes. A node i may appear more than once in the tree because i can be in more than one loop-free path. For example, the optimal-solution tree for the example in Figure 2 is shown in Figure 4 (including both solid lines and dotted lines).

3

Algorithm 2 The optimal solution

1: return DFSOptimal(s);

DFSOptimal(i)

1: **if** $i = d$ **then**
2:    return v as opu_i;
3: **for** each neighbor j not in the path from s to i **do**
4:    obtain opu_j, relay set, and order through DFSOptimal(j);
5: return opu_i, relay set, and order through OpUtility(i);

benefit	opu_d	$opu_1(L/R)$	$opu_2(L/R)$	opu_s
$v = 20$	20	15 / 16.1	16.1 / 15	11.8
$v = 30$	30	24 / 25.7	25.7 / 24	21.2

benefit	opu_d	opu_1	opu_2	opu_s
$v = 20$	20	15	16.1	11.6
$v = 30$	30	24	25.7	20.9

TABLE I
TOP AND BOTTOM TABLES ARE OPRENUS OF NODES IN FIGURE 2 BY THE OPTIMAL AND THE RENU-GUIDED SOLUTIONS, RESPECTIVELY.

As mentioned in Section III-A, once a node's downstream neighbors and their corresponding OpRENUs are known, the OpRENU of the node can be calculated through Algorithm 1. For a node in the optimal-solution tree, its downstream neighbors are its children nodes because they will not incur loops. Initially, only the destination's OpRENU is known, which is v. Therefore, relay selection and prioritization for all nodes can be executed in a bottom-up way (from destination up to source) through the optimal-solution tree.

The formal description of this process is presented in Algorithm 2. Starting from source s, Algorithm 2 recursively explores the current node's downstream neighbors (neighbors not appearing in the path from source to the current node). Once the current node becomes the destination or all its downstream neighbors return their OpRENUs, the OpRENU of the current node, the corresponding cost level, the relay set, and the relay priorities can be obtained through Algorithm 1. In the actual routing process, an intermediate node selects an appropriate downstream neighbor set based on its position in the optimal-solution tree derived from the piggybacked routing history (the path from s to the node).

To complete the example in Figure 4, the top table in Table I gives the corresponding OpRENU for each node in the tree. In the table, $opu_1(L/R)$ denotes node 1's OpRENU in the left/right subtree, since the different positions of a same node in the tree may have different downstream neighbors and different OpRENUs. For the left subtree, if $v = 20$ then $opu_1 = 20 \cdot 0.9 - 3 = 15$ and $opu_d = 20$. According to Algorithm 1 (OpUtility), node 2 in the left subtree selects d and node 1 as downstream neighbors while node 2 in the right subtree selects node 1 as the downstream neighbor. Other nodes' OpRENUs can be calculated similarly.

Despite its optimality, the complexity of the optimal solution is very high. Suppose we have n nodes in the network, the maximum number of paths is the sum of the permutations of the combinations of different numbers of intermediate nodes in the path: $\sum_{k=1}^{n-2}(C_{n-2}^k \cdot P_k^k) = \sum_{k=1}^{n-2} \frac{(n-2)!}{(n-2-k)!} = \omega(2^n)$, where ω stands for lower bound. It is not an efficient solution for real applications. Therefore, we consider a heuristic solution that uses the RENU of each node as guidance to determine downstream neighbors and calculate their corresponding OpRENUs.

C. The RENU-Guided Solution

The fundamental reason for the high complexity of the optimal solution is that there is no total order among nodes so that a node's downstream neighbors can be uniquely determined by a single metric, such as the ENT of ExOR [4]. In this section, we propose a heuristic solution that uses the RENU of each node to determine a total order among all the nodes. For any node, all its neighbors with higher RENUs are its downstream neighbors. By introducing this total order, not only can the downstream neighbors of any node be uniquely determined, but also the OpRENUs of all nodes can be calculated sequentially in the decreasing order of their RENUs. Although downstream neighbors are determined based on RENU, their priority orders are decided based on OpRENUs as will be discussed.

The benefit of the initial ordering is as follows: 1) a node's RENU reflects its closeness to the destination in terms of utility (i.e., how close its utility is to the utility of the destination); 2) with this initial ordering among all nodes, the destination is reachable. Note that a random ordering cannot guarantee the reachability of the destination. For example, in Figure 2, if the ordering is $1 < d < s < 2$, s cannot find a routing scheme to the destination. Note that in the ordering, node i can use node j as its downstream neighbor if and only if $i < j$.

The formal description of our RENU-guided solution is presented in Algorithm 3, which utilizes Algorithm 1 (OpUtility) to calculate a node's OpRENU when its downstream neighbors and their corresponding OpRENUs are determined. In our algorithm, the initial ordering of each node is identified through a method similar to Dijkstra's shortest-path algorithm. We also have an efficient distributed and approximated implementation to calculate both RENU and OpRENU in Section III-E.

Initially, only the destination's RENU and OpRENU are known, both of which are v. All nodes are initially unordered. The order of the destination is the first. All the neighbors of the destination will update their RENUs. Then, the node with the largest RENU among the unordered nodes will be selected as the second-to-highest-order node. This node will regard the destination as its downstream neighbor and update its OpRENU. Its neighbors will update their RENUs accordingly. This process will repeat until the order of the source is determined. Note that relay selection and prioritization of a node can be done once the node's order has been determined, since its downstream neighbors (the neighbors with higher RENUs), and their corresponding OpRENUs have been determined at that time.

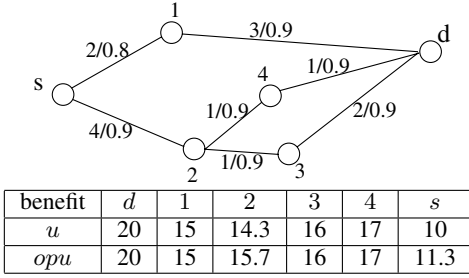| benefit	d	1	2	3	4	s
u	20	15	14.3	16	17	10
opu	20	15	15.7	16	17	11.3

Fig. 5. An example illustrating Algorithm 3.

Algorithm 3 guarantees that nodes are sequentially extracted from the set of unordered nodes based on their RENUs. RENU reflects the closeness to the destination (i.e. utility distance), but we choose OpRENU instead of RENU to determine the priority of the nodes in the relay set in Algorithm 1 (OpUtility). There are two reasons that both of these metrics are needed in our algorithms. First, prioritizing node i's relay set by OpRENU instead of RENU leads to a larger OpRENU of a node i, which enables a better routing scheme for node i. Second, using RENU instead of OpRENU (which has no initial total order) to reflect the utility distance will avoid loops and greatly reduce the complexity by avoiding the need to enumerate all possible paths.

Figure 4 (with only solid lines) shows the "trimmed" solution tree for the RENU-guided solution. Here the RENU order is $s < 2 < 1 < d$. A node's different positions in the tree have the same children nodes, and there is no need to piggyback routing history in the actual forwarding. The bottom table in Table I records the OpRENU results of the nodes in Figure 2. Comparing the two tables in Table I, we can see that OpRENUs generated by this heuristic algorithm only deviate very slightly from the optimal result.

D. Example

To illustrate the heuristic algorithm, we give an example as shown in Figure 5, where the benefit v is set to 20. Initially, node d is selected. Its three neighbors 1, 3, and 4 are relaxed, with their RENUs updated from 0 to 15, 16, and 17, respectively. At present, nodes 1, 3, and 4 have only one relay, so their RENUs are equal to their OpRENUs. d itself is added into the downstream neighbors of 1, 3, and 4. Then, node 4 (with the second largest RENU) is added to the set of ordered nodes. Node 4's OpRENU is 17 (equal to its RENU), its transmission cost is 1, and its downstream node is d. Sequentially, nodes 3, 1, 2 and s are selected. When it is time for 2 to be selected, 2 has two downstream neighbors (3 and 4). 2 has only one cost level, which is 1. $opu_2 = opu_4 \times 0.9 + opu_3 \times 0.9 \times (1 - 0.9) - 1 = 15.74$, which is larger than opu_1, which is 15. Finally, when s is selected, s has two downstream neighbors (1 and 2). s's cost level set is $\{2, 4\}$. s sets its transmission cost to 4 and determines its relay set $\{1, 2\}$ because the expected network utility that is calculated based on the relay set associated with cost level 4 is larger than that of cost level 2. The OpRENU and RENU of each node are listed in the table under Figure 5.

Algorithm 3 The RENU-guided solution

1: Initialize opu_d and u_d to v;
2: **while** s's order is not determined **do**
3:     Remove node i with maximal u_i from the set of unordered nodes;
4:     Add i into the set of ordered nodes;
5:     **if** $i \neq d$ **then**
6:         Call OpUtility(i);
7:     For each unordered neighbor j, Relax(i, j);

Relax(i, j): Calculate u_j from u_i

1: **if** $u_j < u_i \cdot p_{i,j} - c_{i,j}$ **then**
2:     Update u_j: $u_j \leftarrow (u_i \cdot p_{i,j} - c_{i,j})$;
3:     Record node i as j's downstream neighbors;

E. Distributed Implementations

We consider the distributed implementations for both the optimal and heuristic solutions. The one for the optimal solution can achieve the optimal result, but it is still costly. The other one for the RENU-guided solution is an approximation with some desirable features that uses a back-off mechanism.

The optimal solution. In the distributed implementation of the optimal solution, we adopt a link-state-based protocol. Initially, each node broadcasts its local network view (link cost and link reliability associated with the node) to every other node. At the end of this, each node has a global view of the network. There are three major steps: 1) Each node broadcasts its local link state to other nodes, and builds a global view of the network based on the received link state information. 2) Each node constructs the optimal-solution tree after exhaustive searching of all the paths from source to destination, finding the OpRENU for each node in the tree. 3) Upon receiving a packet during the routing process, a node applies the normal OR process to determine whether the node is the forwarder. 4) If the node should forward the packet, it examines the path that the packet has traversed to find the corresponding position in the tree. The node decides its relay set and the order of the relay set according to this position in the tree.

The RENU-guided solution. In the above implementation, each node needs to collect global link state information and the complexity of building the optimal-solution tree is high. We also develop a distributed implementation of the RENU-guided solution, which can be gracefully integrated into a reactive routing protocol, such as AODV [1] or DSR [2].

In the distributed implementation of the RENU-guided solution, RENU and OpRENU could be treated as the summary of local link state information. Each node need not propagate all available local link state information to its upstream node. Instead, it propagates summarized routing information, RENU and OpRENU, to its upstream nodes. Each node locally determines its relay set and the priorities of nodes in the relay set according to the summarized routing information. The following process focuses only on the computation of RENU and OpRENU of each node. 1) The source sends out a message to inform the destination of its benefit (v). 2) The destination
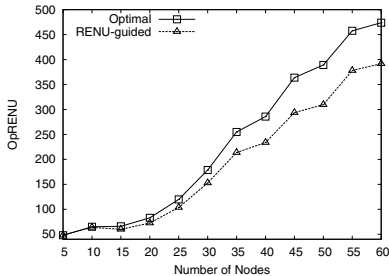
Fig. 6. The comparison between the optimal solution and the RENU-guided solution.

broadcasts its RENU (and OpRENU) to initialize a priority discovery phase that will form a globally directed flooding tree rooted at the destination. 3) Upon receiving the first RENU, each node i, including the source, calculates it initial RENU (i.e., u_i) and sets a timer w_i which is proportional to the utility distance and time period since the original message was sent out at the destination. 4) Before timeout, each node improves its RENU based on the received RENUs (and OpRENUs) of its neighbors, adjusts its timer, and adds the nodes from which it receives RENUs into its downstream node set. 5) After timeout, each node computes its OpRENU based on Algorithm 1, and sends out its RENU and OpRENU to all its neighbors.

The initial value of the timer is a function in which w_j is proportional to $v - u_i$ (utility distance) and is adjusted by $t^* - t$ (time period), where t^* is the current time and t is the time the message was sent out at the destination. We also assume the initial time t is attached when the destination initiates the priority discovery phase. The higher the u_i, the shorter time node i will back-off before it sends out its priority. Whenever a node j receives messages from its neighbors that improve its RENU, it will reduce the remaining back-off time accordingly. Once the timer is properly set, the node with the maximum RENU of all the unordered nodes will send out its reply first, which includes its RENU and OpRENU. Each node's downstream neighbors will include all neighbors with higher RENUs. So the best transmission cost, relay set and the order of the relay set can be decided locally at each node. Although the distributed implementation is an approximation, it has two desirable features. First, the calculation is distributed and each node decides its own transmission cost and relay set. Second, it greatly reduces the transmission overhead, as only the RENU and OpRENU which summarize the link state information will be propagated.

IV. SIMULATION

In this section, we give an evaluation of our metrics and solutions. Without loss of generality, the link cost is modeled as the energy consumption. We compare different metrics for determining the priority of the nodes in the relay set under the framework of opportunistic routing. The metrics include: (1) minimum hop count, (2) maximum path reliability (path delivery ratio), (3) OpRENU, and (4) minimum cost.

A. Simulation Environment

We set up the simulation in a $900m \times 900m$ area, which is the target field. We assume nodes are homogeneous and can be deployed in this area arbitrarily. We fix the position of source s and destination d at locations $(50m, 450m)$ and $(850m, 450m)$, respectively, and randomly deploy the intermediate nodes. The cost of any link (i, j) is generated according to $c_{i,j} = dist_{ij}^{\gamma}$, where $\gamma = 2$. The stability of any link (i, j) is the function of the received power at node j (denoted as R_j). As in [14], the relation between bit-error-rate (p^{ber}) and received power (R_j) is a function of the modulation scheme. More specifically, we have $p^{ber} \propto erfc(\sqrt{\frac{R_j}{K}})$, where K is the ambient noise (noise spectral density) and $erfc(x)$ is defined as $erfc(x) = 1 - \int_0^x e^{-t^2} dt$. Since we are interested in the general dependence of the error rate on the received power as opposed to the details of a specific modulation scheme, we assume that the packet-error-rate p is equal to the multiplication of p^{ber} and the packet size. The received signal power is equal to the transmission power divided by $dist^{\gamma}$. In the simulation, the movement of nodes is characterized by the link reliability of nodes.

For each set of parameters, we run each algorithm 100 times and use the average value of the results to evaluate the performance. In the simulation, we consider n, the number of nodes, as the tunable parameter. We set the maximum transmission range to $300m$. Since the transmission range is fixed, the connectivity of the network is controlled solely by the value of n. We also vary the benefit value from 1000 to 3000.

B. Simulation Results

We first compare our RENU-guided solution with the optimal solution. Because the optimal solution is an exponential algorithm in terms of network size, and it is very slow in the NS-2 simulator, we conduct this experiment in our customized simulator. The rest of the simulations are conducted in NS-2 (version 2.29). The result is presented in Figure 6. We set reliability range: $\alpha = 0$ and $\beta = 1$. The benefit value is set to 1000. We vary the number of nodes from 5 to 60 in increments of 5. The simulation results show that our RENU-guided solution is close to the optimal solution. This illustrates that RENU provides good guidance for relay selection and prioritization. The reason for this is that RENU characterizes those nodes that contribute the most to the increment of OpRENU.

We then conduct sensitivity analysis, i.e., evaluate the effect of parameters such as ambient noise K and benefit value v. As discussed above, the stability of link (i, j) is generated based on the received power R_j. As a comparison, we also randomly generate the stability from $[0, 1]$. We use *Dependent Stability* to denote the former and *Independent Stability* to denote the latter. We consider three values of ambient noise: $K = 1, 2$, and 3. The simulation result is illustrated in Figure 7(a). As expected, the ambient noise decreases the stability, and thus reduces the expected utility. Moreover, we observe that the effects of the ambient noise on the expected utility have similar trends for different values of ambient noise. Therefore, we
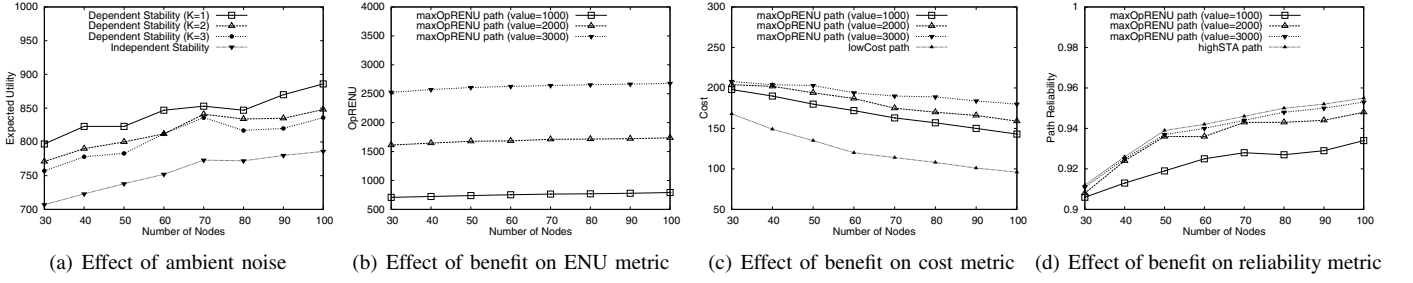
| (a) Effect of ambient noise | (b) Effect of benefit on ENU metric | (c) Effect of benefit on cost metric | (d) Effect of benefit on reliability metric |

Fig. 7. The effect of benefit on (a) utility, (b) cost, and (c) reliability.

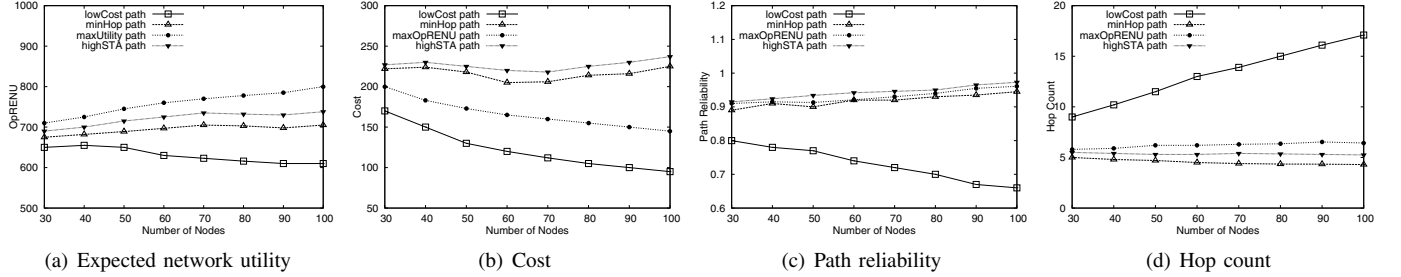| (a) Expected network utility | (b) Cost | (c) Path reliability | (d) Hop count |

Fig. 8. Performance comparison of four optimal routing paths under four different metrics.

set $K = 1$ thereafter in the remainder of the simulations. We also observe that the maximum expected path, based on the stability generated according to the received power, has a larger expected utility than that based on randomly-generated stability. The reason for this is that the former has a smaller stability range compared with the latter.

We also evaluate the effect of the value of benefit on the computation of the optimal route. Roughly speaking, a source with higher benefit is more likely to avoid taking risks by selecting a low cost OR scheme that is less reliable. If the value of v reflects the priority of a routing task, a higher priority routing task should select a more stable and probably a more costly OR scheme. Figures 7 (b)-(d) verify our claim. These figures compare the OpRENU, cost, and reliability of the OR scheme with three different benefit values: 1000, 2000, and 3000. With no surprise, Figure 7 (b) shows that the utility is proportional to the benefit under the same network setting. In Figure 7 (c), the lowest-cost OR scheme is used as a low bound for the maxOpRENU OR scheme under different benefit values because the cost of maxOpRENU OR scheme cannot be less than that of the lowest-cost OR scheme. Similarly, in Figure 7 (d), the most reliable OR scheme is used as an upper bound. The simulation results show that the benefit value has a direct impact on the OpRENU, cost, and reliability of OR schemes, but it does not change the trend. Hence, in the rest of the simulation, we simply set the benefit value to 1000.

To illustrate that the OpRENU can be used to efficiently allocate energy cost over networks, we compare it with three other metrics (minimum hop count, lowest cost, and highest reliability). For each metric, we compute the corresponding OR scheme. The four OR schemes are compared using different metrics. In Figure 8, the OR schemes under maximum

OpRENU based on Algorithm 3, minimum hop count, lowest cost, and highest reliability are abbreviated as maxOpRENU, minHop, lowCost, highSTA, respectively.

Figures 8 (a)-(d) compare the four optimal paths under the OpRENU metric, cost metric, path reliability metric, and hop count metric, respectively. Figure 8 (a) shows that the maxOpRENU-based OR scheme has the best performance in terms of expected utility. From Figures 8 (b) and (c), we can see that the maxOpRENU's performance is the second best in terms of cost and path reliability. The results show that our OpRENU metric is useful for evaluating routing performance in multi-hop wireless networks. The maxOpRENU achieves a good trade-off between cost and reliability.

In Figures 8 (a)-(d), the minimal-hop-count OR scheme yields similar performance results to the most reliable path. The reliability of a path is equal to the product of the reliability of all links on the path. Because the link reliability is uniformly distributed, the lower the hop count is, the higher the path reliability will be. Figures 8 (c) and (d) verify the relation between hop count and path reliability.

In Figure 8 (a), in terms of OpRENU, the highSTA OR scheme has better performance than the lowest-cost OR scheme. The reason is that reliability has a greater effect on OpRENU than on the cost. In Formula (2), if benefit v is large enough, the OpRENU will decrease by half with the reliability P_R decreasing by half. However, the OpRENU will not decrease too much with the cost C_R doubled. Figure 8 (a) shows that the OpRENUs of the maxOpRENU OR scheme, the most reliable OR scheme, and the minimal hop count path increases as the number of nodes increases, but with the OpRENU of the lowest-cost OR scheme, it decreases instead. With more nodes, more paths are available. As the number of nodes increases, although the lowest-cost algorithm

has more choices, the selected OR scheme will have more hop counts, which usually incur lower reliability. The impact of cost decrement cannot make up for the impact of the decrement in reliability.

In Figure 8 (b), both the maxOpRENU OR scheme and the lowest-cost OR scheme decrease with the increment of the number of nodes, but the most reliable OR scheme and the minimal-hop-count OR scheme do not. As we have argued in Figure 8 (a), with the increment of node numbers, the available paths increase, and hence lowest-cost paths will be available. However, the cost of the most reliable path and the minimal hop count path do not necessarily decrease. Our maxOpRENU adequately balances the trade-off between reliability and cost.

In Figure 8 (c), except for the lowCost path, the path reliability increases with the increment of the number of nodes. We have discussed the reason for this in the argument for Figure 8 (a). The desirable result is that the maxOpRENU OR scheme shows good path reliability. The lowest-cost OR scheme is the worst of the four algorithms in terms of OpRENU. The result is not surprising because the path reliability is equal to the multiplication of the link reliability and hence the link reliability has a great influence on OpRENU. The above experiment illustrates that OpRENU is an efficient metric to assess the utilization of network resource.

The simulation results can be summarized as: 1) The RENU-guided solution is a good approximation to the optimal solution. 2) The OpRENU is sensitive to the range of reliability, the number of nodes, the transmission range. 3) The OR scheme based on OpRENU has better performance compared with three other metrics (minimum hop count, lowest cost, and highest reliability). 4) Our OpRENU metric adequately balances the trade-off between the reliability and cost.

V. RELATED WORK

Opportunistic routing (OR) [4], [12], [13] is mainly proposed to address the unreliable communication in multi-hop wireless networks by utilizing the broadcast advantage of wireless communication. Relay selection and prioritization are two key issues of OR protocol design. Some works [8], [12], [17] apply the OR scheme to geographic routing, and select nodes that are geographically closer to the destination to form a relay set. Other OR protocols [4], [5], [7] select any node with smaller cost to the destination as a relay. In the model proposed in this paper, relays do not need to be "closer" to the destination.

Numerous works adopt network utility as the optimization objective. Li, Xue, and Nahrstedt [9] present a price-based scheme to effectively allocate resources among multiple multi-hop flows. Their approach maximizes the aggregated utility of flows (the network utility), while maintaining basic fairness among multiple flows. Qiu and Marbach [11] propose a market-based (similar to utility-based) approach to efficiently allocate bandwidth in multi-hop wireless networks. In our previous work [10], we propose the basic utility-based model. Although there are numerous existing works applying the network utility related approaches, as far as we know, none of them combine reliability and link cost to design an optimization model that will maximize the expected network utility.

VI. CONCLUSION

In this paper, we explore the optimality of OR for a utility-based routing. We observe that the optimal scheme requires exhaustive searching of all paths from source to destination. We propose both optimal and heuristic solutions to select relays and determine priorities among them, and implement both solutions in a distributed way. Extensive simulations on NS-2 and our customized simulator are conducted to verify the effectiveness of the heuristic approach compared with the optimal one. In the future, we will explore multi-path forwarding under the OR framework by integrating network coding techniques. Also, we will study the impact of retransmission in utility-based routing, which may or may not increase utility depending on the success of eventual packet delivery.

REFERENCES

[1] C. Perkins. Ad Hoc On Demand Distance Vector (AODV) Routing IETF, Internet Draft, draft-ietf-manet-aodv-00.txt, November 1997.
[2] D. B. Johnson and D. A. Maltz. Dynamic Source Routing in Ad hoc Wireless Networks. Mobile Computing, 1994.
[3] D. Aguayo, J. Bicket, S. Biswas, G. Judd, and R. Morris. Link-level measurements from an 802.11b mesh network. In *Proceedings of ACM SIGCOMM, 2004*, 2004.
[4] S. Biswas and R. Morris. ExOR: Opportunistic multi-hop routing for wireless networks. In *Proceedigs of ACM SIGCOMM'05*, pages 133–144, 2005.
[5] S. Chachulski, M. Jennings, Sachin K., and D. Katabi. Trading structure for randomness in wireless opportunistic routing. In *Proceedings of ACM SIGCOMM'07*, 2007.
[6] T. Cui, L. Chen, T. Ho, and S. Low. Opportunistic source coding for data gathering in wireless sensor networks. In *Proceedings of the 4th IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS'07)*, 2007.
[7] H. Dubois-Ferriere, M. Grossglauser, and M. Vetterli. Least-Cost Opportunistic Routing. In *Proceedings of 2007 Allerton Conference on Communication, Control, and Computing*.
[8] H. Füßler, J. Widmer, M. Käsemann, M. Mauve, and H. Hartenstein. Contention-based forwarding for mobile ad hoc networks. *Elsevier's Ad Hoc Networks*, 1(4):351–369, 2003.
[9] B. Li, Y. Xue, and K. Nahrstedt. Price-based resource allocation in wireless ad hoc networks. Technical report, UIUCDCS-R-2003-2331, Univ. of Illinios at Urbana-Champaign, 2003.
[10] M. Lu and J. Wu. Social welfare based routing in ad hoc networks. In *Proceedings of the 35th International Conference on Parallel Processing (ICPP'06)*, pages 211–218, 2006.
[11] Y. Qiu and P. Marbach. Bandwidth allocation in wireless ad hoc networks: A price-based approach. In *Proceedings of IEEE INFOCOM'03*, 2003.
[12] R. C. Shah, A. Bonivento, D. Petrovic, E. Lin, J. van Greunen, and J. Rabaey. Joint optimization of a protocol stack for sensor networks. In *Proceedings of IEEE Military Communications Conference, 2004 (MILCOM'04)*, pages 480–486, 2004.
[13] R. C. Shah, S. Wietholter, A. Wolisz, and J. M. Rabaey. When does opportunistic routing make sense? In *Proceedings of the Third IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOMW'05)*, 2005.
[14] B. Sklar. *Digital Communications, 2nd ed*. Prentice-Hall, Inc., 2001.
[15] Y. Wang, M. Martonosi, and L. Peh. A new scheme on link quality prediction and its applications to metric-based routing. In *Proceedings of ACM SENSYS'05*, pages 288–289, New York, NY, USA, 2005.
[16] S. Zhong, L. Li, Y. G. Liu, and Y. Yang. On designing incentive-compatible routing and forwarding protocols in wireless ad-hoc networks: an integrated approach using game theoretical and cryptographic techniques. In *Proceedings of ACM MOBICOM'05*, pages 117–131, 2005.
[17] M. Zorzi and R. Rao. Geographic random forwarding (GeRaF) for ad hoc and sensor networks: energy and latency performance. *IEEE Transaction on Mobile Computing*, 2(4), 2003.