Optimal Forwarder List Selection in Opportunistic Routing

Yanhua Li [*], Wei Chen [†], and Zhi-Li Zhang[*]

[*]University of Minnesota, Minneapolis, USA
[†]University of Electronic Science and Technology of China, Chengdu, China

***Abstract*—Unlike traditional wireless routing protocols which use a single fixed path, *opportunistic routing* explicitly takes advantage of the broadcast nature of wireless communications by using a set of forwarders to *opportunistically* perform packet forwarding. A key issue in the design of opportunistic routing protocols is the *forwarder list selection* problem. In this paper we establish a general theory for analyzing the forwarder list selection problem, and develop an optimal solution, the *minimum transmission selection (MTS)* algorithm, which minimizes the expected number of transmissions and it can be incorporated into existing opportunistic routing protocols to select optimal forwarder lists. Our theory and algorithm can also be generalized to optimize other routing objectives such as minimizing the expected transmission time or energy consumption in opportunistic routing. Through extensive simulations, we demonstrate that in more than 90% cases the MTS algorithm outperforms the ETX forwarder selection scheme used in existing opportunistic routing protocols such as ExOR and MORE.**

***Index Terms*—Forwarder list, Wireless routing, Opportunistic routing, Dynamic programming**

I. Introduction

The *opportunistic routing* paradigm (see, e.g., [3], [5]) has opened a new avenue for designing routing protocols in multi-hop wireless networks. Unlike traditional routing wireless protocols such as DSR, AODV [8], [14] which rely on a (pre-selected) single *fixed* path for delivering packets from a source to a destination, opportunistic routing explicitly takes advantage of the broadcast nature of wireless communications to allow multiple (pre-selected) forwarders to *opportunistically* deliver packets from a source to a destination. The path a packet takes depends on which forwarders happen to receive it, and is thus non-deterministic. As a result, opportunistic routing can better cope with the *lossy, unreliable* and *varying* link qualities that are typical of wireless networks.

While the basic idea of opportunistic routing is fairly straightforward, the forwarder list selection problem is the key issue in designing an opportunistic routing protocol. In order to fully realize the benefits of opportunistic routing, a list of forwarders must be judiciously selected, and these forwarders must *co-ordinate* their packet forwarding—either explicitly or implicitly—in order to successfully and efficiently deliver packets from the source to the destination.

Most of existing studies [3], [4], [9], [12], [13] on opportunistic routing use *ETX* as routing metric to select forwarder list or assume the optimal forwarder list is pre-selected. We refer to these ETX based opportunistic routing protocols collectively as *ETX-OR*. ETX-OR ranks the potential forwarders using their *minimum path ETX*, namely, the sum of the constituent link ETX's along the "best path" (in terms of path ETX) from a node to the destination. We see that ETX-OR in fact utilizes a metric defined on a *single fixed path* for forwarder list selection and ordering, thereby ignoring the very effect of opportunistic routing. As first pointed out in [6], [16] (and shown via an explicit example in this paper), such a forwarder list selection method is in general *sub-optimal*!

This paper is devoted to addressing the fundamental problem of *forwarder list selection* in opportunistic routing. We establish a general *event-based analytical methodology* for studying the forwarder list selection problem, and develop an *optimal* solution which minimizes the expected number of transmissions under an ideal setting—the perfect ACK assumption (see section II). In section III we develop the general theory and present an optimal solution, referred to as the *minimum transmission selection (MTS)* algorithm, for solving the forwarder list selection problem. Given a general topology, the MTS algorithm selects and computes the *optimal forwarder list* for *any* node to a given destination that minimizes the total expected number of transmissions, using a dynamic programming formulation. In section IV we discuss how to relax the perfect ACK assumption by proposing two heuristics to address the unreliable, asymmetric link qualities, thereby dealing with the effect of imperfect ACKs. In section V, we compare the performances of the MTS algorithm and ETX based forwarder list selection scheme in ExOR and MORE using extensive simulations. The paper is concluded in section VI.

Related Work. ExOR [3] is perhaps one of the first fully implemented, practical opportunistic routing protocols for wireless mesh networks, where the notion of a (prioritized) forwarder list is used for specifying and coordinating intermediate forwarders. MORE [4] introduces a linear network coding strategy to utilize spatial reuse in opportunistic routing, and achieve higher throughput than ExOR. MIXIT [9] improves the throughput further by using network coding at the physical layer. CodeOR [12] also extends MORE by allowing multiple coded batches to be transmitted at the same time. The authors in [15] employ an optimization framework to perform rate control and enable MORE to attain better performance. All of the above routing schemes either select the forwarder list based on the minimum (path) ETX, or assume that an optimal forwarder list is given *a priori*. There are a number of

This work is supported in part by the National Science Foundation grants CNS-0626812, CNS-06268, CRI 0709048 and University of Minnesota DTC DTI grant.

978-1-4244-5113-5/09/$25.00 ©2009 IEEE

other studies and protocols with various opportunistic routing flavors. Due to space limitation, we do not discuss them here.

Zhong *et al.* [16] introduces a new routing metric, EAX (expected anycast transmission), and propose a *heuristic* algorithm for computing a set of candidate forwarders. Inspired by [16], Dubois-Ferriere *et al.* [6] introduces the notion of anycast link cost (ALC) associated with a set of forwarders, and uses this notion to define a (somewhat abtsruse) cost function associated with a forwarder list (referred as "opportunistic route"). Based on this cost function, the authors propose the LCOR (least-cost opportunistic routing) algorithm, which essentially enumerates all the possible neighboring node combinations (in a recursive manner) to compute the best opportunistic route with the least cost. Due to its potentially exponential time complexity, heuristic policies have to be incorporated in LCOR. Concurrent to our work, [10] refines the least cost function used in [6], and develops a similar Dijkstra-like algorithm to find the least cost (interpreted as some sort of "hyperlink distance") opportunistic route. In both [6] and [10], the cost of an opportunistic route (or forwarder list) is given as a *definition*, the physical meaning of which is not entirely clear. Hence the optimality of the algorithms is established with respect to such an (abstract) cost function. Due to this reason, neither paper addresses the impact of *imperfect* ACKs on the forwarder list selection in opportunistic routing.

In contrast, we formulate the forwarder list selection problem with the *explicit* goal of minimizing the expected number of transmissions. Under the (ideal) perfect ACK assumption, we derive a routing metric (cost function) associated with a forwarder list, which is precisely the expected number of transmissions when the given forwarder list is used for opportunistic routing, and thus our optimal algorithm selects the forwarder list that minimizes the total expected number of transmissions. Our methodology and algorithm can be generalized to account for and optimize other (physically meaningful) routing costs or objectives such as expected transmission time or expected energy consumption.

II. Background and Problem Setting

As in ETX-OR, we assume that a *global prioritized forwarder list* is used for opportunistic packet forwarding, and the same (batch mode-based, round-by-round, prioritized) *packet forwarding mechanisms* are employed, which is described below. Let $\{s, u_m, u_{m-1}, \ldots, u_1, d\}$ be an (ordered) forwarder list, where s is the source, d is the destination, and u_1,...,u_m are a set of m (intermediate) forwarders. The nodes are ordered based on their increasing priorities from left to right: namely, the destination node d has the highest priority, s the lowest priority, and for $1 \leq i < j \leq m$, u_i has higher priority over u_j. (If $m = 0$, no intermediate forwarder is used.)

We will show that the forwarder list selection scheme used in ETX-OR is *sub-optimal* through a simple example. Recall that ETX-OR uses the following two rules to select and order forwarders: *a) candidate set selection rule*: node with a smaller path ETX value (to the destination) than the source will be chosen in the forwarder list; and *b) ordering rule*: forwarders are ordered by their path ETX values to the destination, where a forwarder with smaller path ETX value has higher priority. To show that the forwarder list selected based on these two rules are *sub-optimal*, we consider a simple example shown in Fig.1, where node s is the source, d is the destination, and the number on each link indicates its (symmetric) packet delivery probability. Computing the path ETX using the best path, we see that the path ETX for node v_2 (to d) is $1/0.8 = 1.25$, for node v_1 it is $1/0.45 = 2.22$, and for the source node s it is $1/0.5 = 2$. Hence the forwarder list selected by ETX-OR is $\{s, v_2, d\}$. However, it is not hard to argue that the forwarder list $\{s, v_1, v_2, d\}$ is in fact a better choice, as it is likely to reduce the total number of transmissions to deliver packets from s to d. This is because that v_1 has two opportunistic paths, $v_1 \rightarrow d$ and $v_1 \rightarrow v_2 \rightarrow d$, which together yield a higher packet probability than using either path alone. The expected number of transmissions from v_1 to d using both paths is roughly 1.742, which is far smaller than the path ETX of v_1 (2.22) and is also smaller than the path ETX of s (2). The *sub-optimality* of the ETX forwarder list selection scheme lies in that it uses the *single path* ETX metric for selecting forwarders, and thus fails to account for the *opportunistic* nature of packet forwarding.

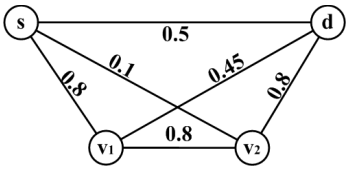

Fig. 1. An example showing that ETX based forwarder list selection scheme is sub-optimal.

In analyzing the forwarder list selection problem in this paper, we therefore employ the *expected number of transmissions* (excluding duplicate transmissions due to unreliable overhearing) as the key metric to compare forwarder lists, and address the two basic questions: *i) which forwarders to select*, and *ii) how to order the forwarders selected*? To simplify our analysis and separate the effect of duplicate transmissions due to unreliable overhearing, we introduce and consider an *ideal setting* with the *perfect ACK* assumption, namely, we assume that lower priority nodes can always overhear the transmissions of higher priority nodes, thus there will be no unnecessary duplicate transmissions. Under this assumption, in this paper we develop a general methodology for analyzing the forwarder list selection problem, and develop a dynamic programming algorithm for finding the *optimal* forwarder list which minimizes the expected number of transmissions for a packet to traverse opportunistically from a source to a destination. We will relax this perfect ACK assumption in section IV by introducing two mechanisms to handle unreliable and asymmetric packet delivery probabilities.

III. Forwarder List Selection: General Theory

In this section, we study the forwarder list selection problem. Under the perfect ACK assumption, we establish a

general theory for solving the general forwarder list selection problem in designing opportunistic routing protocol.

We'll present the MTS algorithm for finding the *optimal* forwarder list which provably minimizes the expected number of transmissions needed to transmit a packet from the source to the destination (under the perfect ACK assumption). We start by establishing a recurrent formula for computing the expected number of transmissions for a given forwarder list.

A. Expected No. of Transmissions for a Given Forwarder List

Consider a general topology,where we have a source s and a destination d, and n intermediate nodes that are arbitrarily connected among themselves (and with s and d).We use z to denote the packet probability from the source node s directly to the destination node d, x_i the packet probability from node s to the (intermediate) node v_i, y_i the packet probability from node v_i to node d, and h_{ij} the packet probability delivery probability from node v_i to node v_j, where $0 \leq z \leq 1$, $0 \leq x_i \leq 1$, $0 \leq y_i \leq 1$ and $0 \leq h_{ij} \leq 1$ [1]. Given a total of n intermediate nodes, we have a total of $\sum_{i=0}^{n} \frac{n!}{(n-i)!}$ possible forwarder lists (including $\{s, d\}$–the the default forwarder list *without using any intermediate node*). Of these $\sum_{i=0}^{n} \frac{n!}{(n-i)!}$ forwarder lists, which one is the best and under what condition? Instead of directly answering this question, we start by establishing a *recurrent* formula for computing the expected number of transmissions for a *given* forwarder list.

For $m = 1, \ldots, n$, let $u_m, \ldots, u_1$ denote an arbitrary (ordered) list of m intermediates nodes taken from the set of n intermediate nodes $\{v_1, v_2, \ldots, v_n\}$. Then $\{s, u_m, \ldots, u_1, d\}$ is a possible forwarder list. We would like to derive a formula to compute the expected number of transmissions to reach destination d using this particular forwarder list. We denote this number by $N_{s,u_m,\ldots,u_1,d}$. More generally, for $k = 1, \ldots, m$, we use $N_{u_k,\ldots,u_1,d}$ to denote the expected number of transmissions to reach destination d *when the packet has reached node* u_k*, but not* u_j*,* $j = 1, \ldots, k-1$. In other words, $N_{u_k,\ldots,u_1,d}$ is the expected number of transmissions to reach destination d *if* u_k is the source node and $\{u_k, \ldots, u_1, d\}$ is the forwarder list. Given these notations, we have the following recurrent formula for computing $N_{s,u_m,\ldots,u_1,d}$.

THEOREM 1 (Recurrent Formula for Expected Number of Transmissions). *Consider a general topology with n intermediate nodes. For $1 \leq m \leq n$, let $\{s, u_m, \ldots, u_1, d\}$ be an arbitrary forwarder list with m (ordered) intermediate forwarders, $u_i \in \{v_1, \ldots, v_n\}$. Then using this forwarder list, the expected number of transmissions for transmitting a packet currently at the source node s to reach the destination node d is given by the following recurrent formula (under the perfect ACK assumption):*

$$N_{s,u_m,\ldots,u_1,d} = \frac{1 + \sum_{k=1}^{m} x_k \prod_{j=1}^{k-1}(1-x_j)(1-z) N_{u_k,\ldots,u_1,d}}{1 - \prod_{k=1}^{m}(1-x_k)(1-z)}. \quad (1)$$

[1]Note that some of these packet delivery probabilities can be 0, which means that the source and destination can be multi-hops away from each other.

Proof: The theorem follows easily by using the following event-based analysis. We use $\mathcal{E}^m := u_m \vee \cdots \vee u_1 \vee d$ to denote the event that when s transmits the packet, either the destination node d or (at least) one of the m intermediate forwarders receives it, whereas we use $\bar{\mathcal{E}}^m (:= \bar{u}_m \wedge \cdots \wedge \bar{u}_1 \wedge \bar{d})$ to denote the complement of this event, namely, when s transmits the packet, neither the destination node d nor any of the m intermediate forwarders receives it, For $1 \leq k \leq m$, let $\mathcal{E}^{k,\not<k}$ denote the event that when s transmits the packet, u_k receives it but none of the higher priority intermediate forwarder, u_j, $1 \leq j < k$, nor d, receives it, and $(\mathcal{E}^{k,\not<k} | \mathcal{E}^m)$ is this event conditioned on $\mathcal{E}^m$ occurring. Note that $(\mathcal{E}^{k,\not<k} | \mathcal{E}^m)$, $k = 1, \ldots, m$, plus the event $(d|\mathcal{E}^m)$ (i.e., the conditional event that the destination node d receives the packet), are mutually exclusive events, and provide a partition of the event $\mathcal{E}^m$. In other words, $\sum_{k=1}^{m} Pr(\mathcal{E}^{k,\not<k} | \mathcal{E}^m) + Pr(d|\mathcal{E}^m) = Pr(\mathcal{E}^m)$. Let $N_{s \to \mathcal{E}^m}$ denote the expected number of transmissions that the source node s has to perform until $\mathcal{E}^m$ occurs, i.e., at least one of the nodes in $\{u_m, \ldots, u_1, d\}$ receives the packet. Given these notations and recall the definition of $N_{u_k,\ldots,u_1,d}$, we have the following recurrent formula for $N_{s,u_m,\ldots,u_1,d}$:

$$N_{s,u_m,\ldots,u_1,d} = N_{s \to \mathcal{E}^m} + \sum_{k=1}^{m} Pr(\mathcal{E}^{k,\not<k} | \mathcal{E}^m) N_{u_k,\ldots,u_1,d}. \quad (2)$$

Note that if $(d|\mathcal{E}^m)$ occurs, then no additional transmissions are needed. The above recurrent formula states that the expected number of transmissions using the forwarder list $\{s, u_m, \ldots, u_1, d\}$ is the sum of the expected number of transmissions by the source node s until $\mathcal{E}^m$ occurs, and given the event, for $1 \leq k \leq m$, if u_k receives it but none of the higher priority forwarders *and nor* does the destination node d receive it, the expected number of transmissions, $N_{u_k,\ldots,u_1,d}$ from the intermediate forwarder u_k to reach d using the forwarder list $\{u_k, \ldots, u_1, d\}$. Clearly, $N_{s \to \mathcal{E}^m} = 1/Pr(\mathcal{E}^m)$ and $Pr(\mathcal{E}^m) = 1 - \prod_{k=1}^{m}(1-x_k)(1-z)$. For $k = 1, \ldots, m$, $Pr(\mathcal{E}^{k,\not<k} | \mathcal{E}^m) = x_k \prod_{j=1}^{k-1}(1-x_j)(1-z)/Pr(\mathcal{E}^m)$. Plugging these formulas into eq.(2) yields eq.(1). ■

B. Minimum Transmissions Selection Scheme

We are now in a position to present the *Minimum Transmissions Selection(MTS)* algorithm, which computes the optimal forwarder list that minimizes the expected number of transmissions. In fact, given a general topology and the destination d, the MTS algorithm computes the optimal forwarder lists from *all sources* to d, using a *dynamic programming* formulation somewhat analogous to the Dijkstra's algorithm (which computes the shortest paths *from a given source to all destinations*).

Given a general wireless topology, initially let $\mathcal{S}$ be the set of all nodes except for a given destination node d. The MTS algorithm for computing the optimal forwarder list from any source node $v \in \mathcal{S}$ to d is described in pseudo-code in Alg. 1. At each iteration of the algorithm, for any node v in $\mathcal{S}$, $FL^d[v]$ records the (best) forwarder list from v to d discovered so far, and $N^d[v]$ denotes the expected number of

transmissions using the (currently best) forwarder list $FL^d[v]$. During the initialization stage (steps 1-6), for each $v \in \mathcal{S}$, clearly $FL^d[v] := \{v, d\}$ is the currently best forwarder list for v, and $N^d[v] = 1/Pr(v \rightarrow d)$ if the packet delivery probability $Pr(v \rightarrow d)$ is non-zero.

At each subsequent iteration while $\mathcal{S}$ is not empty (steps 9-13), we pick the node $u \in \mathcal{S}$ such that $u := argmin_{v \in \mathcal{S}} N^d[v]$ (step 9), i.e., u is the node in $\mathcal{S}$ with the smallest expected number of transmissions $N^d[u]$, and remove it from $\mathcal{S}$ (step 10). It can be argued (see the next paragraph) that $FL^d[u]$ contains the optimal forwarder list for u with the minimum $N^d[u]$ (among all possible forwarder lists for u to d), and it is therefore removed from $\mathcal{S}$ for further consideration in the future iterations. Given this, we now consider any neighbor node v of u that is still in $\mathcal{S}$ (step 11). If v is a neighbor of u, we *merge* the current best forwarder list $FL^d[v]$ with that of u, $FL^d[u]$, to obtain a new forwarder list for v (step 12). The **merge** operation combines and orders the nodes in $FL^d[v]$ and $FL^d[u]$ (except for v and d) based on the order at which these nodes are removed from the set $\mathcal{S}$: the earlier a node w in $FL^d[u]$ or $FL^d[v]$ is removed from $\mathcal{S}$, the higher the priority of w will be (clearly d has the highest priority, and v the lowest), and the new *merged* forwarder list $FL^d[v]$ is thus of the form $\{v, u, \ldots, d\}$.

We then update $N^d[v]$ with the expected number of transmissions using this new merged forwarder list $FL^d[v]$, computed via Eq.(1) (step 13). This procedure continues until the optimal forwarder list is computed for all nodes (i.e., until $\mathcal{S}$ is empty).

Algorithm 1 The MTS algorithm for computing the optimal forwarder lists of all sources v's to the destination d. ($\mathcal{S}$ is the set of all nodes except d)

1: **//Initialization:**
2: **for** each vertex v in $\mathcal{S}$ **do**
3: **if** $P_r(v \rightarrow d) > 0$ **then**
4: $N^d[v] := 1/P_r(v \rightarrow d)$;$FL^d[v] := \{v, d\}$
5: **else**
6: $N^d[v] := \infty$;$FL^d[v] := \phi$
7: **//Iterations:**
8: **while** $\mathcal{S}$ is not empty **do**
9: $u :=$ node in $\mathcal{S}$ with smallest $N^d[\cdot]$ (i.e., $u := argmin_{w \in \mathcal{S}} N^d[w]$)
10: remove u from $\mathcal{S}$
11: **for** each neighbor v of u and v is in $\mathcal{S}$ **do**
12: $FL^d[v] :=$ **merge**$(FL^d[u], FL^d[v])$
13: $N^d[v] := N^d_{FL^d[v]}$ (where $N^d_{FL^d[v]}$ is computed using eq.(1))
14: **RETURN** $FL^d[.]$ and $N^d[.]$

The optimality of the MTS algorithm can be established by induction and proof-by-contradiction. Due to space limitation, we omit the formal proof here, which can be found in [11].

Extensions. For simplicity of exposition, in this paper we have used *minimizing the expected number of transmissions* as the cost function (or routing metric) in selecting the optimal forwarder list. In fact we can easily account for other routing objectives such as the expected transmission time or expected energy consumption by introducing a *(node-specific) per-transmission cost* c_i at each node i. For example, define $c_i = s/r_i$, where r_i is the transmission rate of node i and s is the packet size, than c_i represents the transmission time of the packet at node i. Likewise, define $c_i = e_i$, then c_i represents the per-transmission energy consumption at node i when transmitting the packet(as defined in [2]). Given a forwarder list $F = \{s, u_m, \ldots, u_1, d\}$, we can then associate a cost function C_F with the forwarder list. Using the same analysis methodology, we can establish a recursive formula to compute this cost function analogous to Theorem 1 as follows:

$$C_{s,u_m,\ldots,u_1,d} = \frac{c_s + \sum_{k=1}^{m} x_k \prod_{j=1}^{k-1}(1-x_j)(1-z)C_{u_k,\ldots,u_1,d}}{1 - \prod_{k=1}^{m}(1-x_k)(1-z)}. \quad (3)$$

Using the above formula as the cost function in Algorithm 1, we can compute the optimal forwarder that optimizes different routing objectives such as the expected transmission time (with $c_i = s/r_i$) and expected energy consumption (with $c_i = e_i$).

IV. Handling Imperfect ACKs

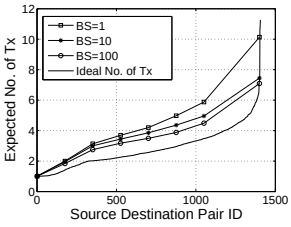

Fig. 2. Batch mode reduces the effect of imperfect ACKs with MTS scheme.

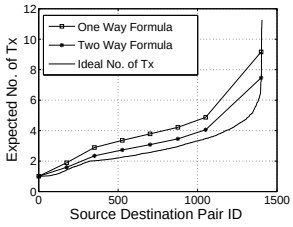

Fig. 3. Effect of two-way link formula 4 (batch size $b = 100$ and $S_{ACK} = 10$).

So far we have developed an optimal forwarder selection algorithm that minimizes the expected total number of transmissions under the perfect ACK assumption. In practice, due to unreliable transmissions (e.g., due to *asymmetric link qualities*), there will be likely *imperfect* ACKs. Such imperfect ACKs would lead to later forwarders no hearing the transmissions of the previous forwarders (or source), resulting in (un-necessary) *duplicate transmissions*.To minimize imperfect ACKs due to unreliable transmissions and asymmetric link qualities, we adopt two heuristic mechanisms, *batch mode* and *two-way link quality formula*, briefly described below. We provide simulation results to show that these two mechanisms together can indeed approximate the perfect ACK assumption fairly well in most scenarios.

Batch Mode. We adopt the batch mode data forwarding mechanism and its associated *(cumulative) bitmap ACK* scheme. Using the batch mode, packets are grouped in a batch of certain size (e.g., $b = 100$ packets), and each packet carries a batch id and packet id (its relative position in the batch) and a *batch map*, where the ith entry of the map contains the id of highest priority node (the destination or a forwarder, based on their relative order in the forwarder list) that has received packet i in the batch; originally it contains the id of the source. When a forwarder broadcasts the packets of a batch it has received, each of them carries the same batch map. Hence upon receiving only one of these packets, a later forwarder (in the forwarder list) will know the status of the packets that have been received by previous forwarders (including the destination) and update its batch map accordingly (please refer

to [3] for details). As a result, the batch mode enhances the probability of overhearing among the forwarders and reduces unnecessary duplicate transmissions.

To show the efficacy of batch mode, especially, the effect of batch size on reducing unnecessary transmissions, Fig. 2 compares the expected total number of transmissions under *perfect ACK* (i.e., assuming all nodes can hear each other's transmissions perfectly) (shown as the *ideal no. of Tx* in the figure) with the (average) number of transmissions actually required with unreliable and asymmetric links using batch mode of varying size. Here the simulations are conducted using the MIT Roofnet dataset (see Section V for more details regarding the simulation settings). For each source-destination pair, the (optimal) forwarder list is computed using the MTS algorithm, and the *average* number of transmissions is computed over 100 simulation runs with different random seeds. The x-axis is the source-destination pair id ordered based on its *ideal number of transmissions.*, and the y-axis is the average or expected of transmissions. From the figure we see that as we increase the batch size, the average number of actual transmissions (which include duplicate transmissions) decreases, and approaches to the ideal number of transmissions with *perfect ACK*. Similar improvements can be observed also when we choose forwarder lists using ETX.

Two-way Link Quality Formula. To deal specifically with asymmetrical link qualities and discourage the use of links with drastic different packet delivery probabilities along its two directions, we introduce the following two-way link quality formula to re-define the packet delivery probability from one node to another and use it in the computation of expected number of transmissions (e.g., as in Eq.(1)):

$$p'_{ij} = p_{ij}[1 - (1 - p_{ji})^{S_{ij}^{ACK}}] \quad (4)$$

where S_{ij}^{ACK} is a parameter that depends on the batch size. In this paper, for batch size b=100, we set $S_{ij}^{ACK} := S^{ACK} = 0.1b = 10$ for all links[2]. This formula takes into account the link qualities of both directions as well as the effect of batch-mode based cumulative ACKs. Comparing two links, one link with the forward link quality p_{12} is 90% and backward link quality p_{21} is 5%, and other link with $p_{13} = p_{31} = 50\%$. Using the two-way link formula, we have $p'_{12} = 36.11\% < p'_{13} = 49.95\%$ (using $S^{ACK} = 10$). In other words, in order to reduce unnecessary duplicate transmissions due to ACK losses, it would be better for node 1 to use node 3 as a higher priority forwarder than node 2. Fig.3 shows the effect of using the two-way link quality formula which discourages the use of bad asymmetric links with our MTS algorithm in ExOR, and as a result, further the number of unnecessary retransmissions. (The same observation also holds when ETX forwarder list selection algorithm is used.) Hence with a relative large batch size (e.g., 100 packets) and the two-way link quality formula, we can reasonably approximate the perfect ACK assumption and perform forwarder list selection accordingly.

[2]Note that the packets within a batch received by a node j (as an intermediate forwarder) will differ, depending on the senders and rounds of transmissions. Hence in general it will be difficult to precisely set S_{ij}^{ACK}. To be fairly conservative, we choose $S_{ij}^{ACK} = 0.1b$ in our study. Alternatively for a fixed source-destination pair and batch size, we could conduct experiments and use estimated average ACK size based on measurements to set S_{ij}^{ACK} for each link. Our simulation study in fact shows that with relatively large batch size, say, b=100, varying S_{ij}^{ACK} does not significantly affect the ordering and selection of forwarder lists in most scenarios.

V. Performance Evaluation and Comparison

We conduct extensive simulations in $ns2$ [7] to evaluate the performances when using our MTS algorithm in ExOR [3] and MORE [4]. The simulation results reported here are based on the MIT Roofnet topology and dataset [1]. There are 38 nodes in the Roofnet topology, and the link quality (or packet delivery probability) between any two nodes is derived from the Roofnet data. The simulation parameters are listed in TableI. There are a total of 1406 source-destination pairs in the Roofnet topology. For each source-destination (in shorthand, src-dst) pair, we run 20 simulations with different random seeds, and the averages of these 20 simulation runs are reported in the discussion below.

We now use different forwarder list selection schemes, MTS and ETX, in ExOR and compare the performances for each src-dst pair, in terms of both the (average) *actual* number of transmissions (including duplicates), as shown in Fig.4, and the throughput (or goodput, i.e., the number of bytes transmitted from the source to the destination per unit time, measured in KB/sec), as shown in Fig.5. In the figures, each dot corresponds to one src-dst pair, where its coordinate (x, y) represent the results under original ETX based ExOR (ETX-ExOR) and MTS-ExOR, respectively. We see that overall the forwarder lists selected by MTS outperform those by ETX: 92.05% src-dst pairs have fewer number of transmissions under MTS than under ETX scheme, and 90.89% src-dst pairs have larger throughputs under MTS than under ETX. There are a small number of pairs for which MTS-ExOR produces poorer performances than ETX-ExOR. Detailed analysis shows that this is mostly due to the effect of imperfect ACKs, causing unnecessary duplicate transmissions. To further analyze and compare the performances of MTS-ExOR and ETX-ExOR, the performance gains are computed using the following formulas: $G_{Tx} = (Tx_{ETX} - Tx_{MTS})/Tx_{ETX}$ and $G_{Thput} = (Thput_{MTS} - Thput_{ETX})/Thput_{ETX}$.From Fig.6, we see that MTS can reduce the number of transmissions by up to 73%, where the upper bound[3] of the reduction is 100%, and increase the throughput by up to 253%.

TABLE I
Simulation Parameters

Parameters	Values
Batch Size & S^{ACK}	100 & 10
Bandwidth	$1Mbps$
Forwarder list selection schemes	MTS, ETX
Routing protocol	ExOR, MORE
Topology	Roofnet
Transmission protocol	UDP
Period of simulation	150s for each simulation
Dataflow time	120s for each simulation

[3]The upper bound of no. of transmissions gain can be easily got from the definition of G_{Tx}

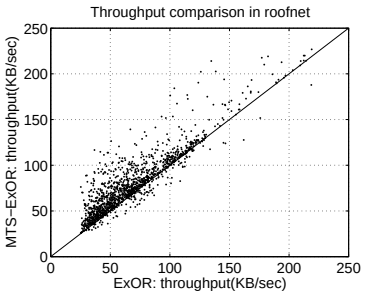

Fig. 4. Throughputs

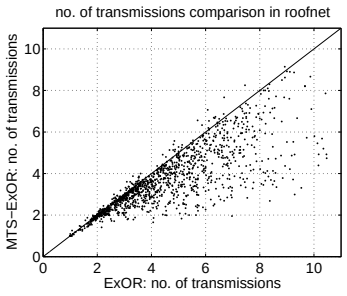

Fig. 5. No. of transmissions

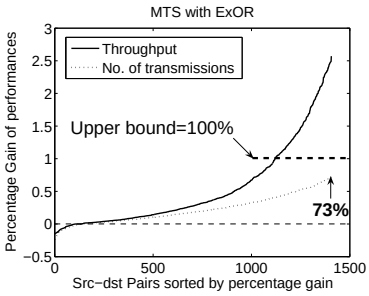

Fig. 6. Percentage gain in ExOR

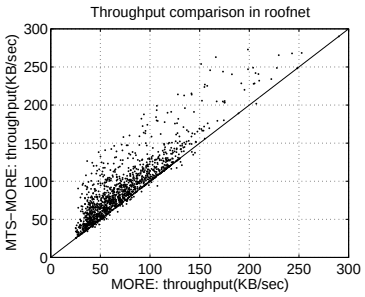

Fig. 7. Throughputs

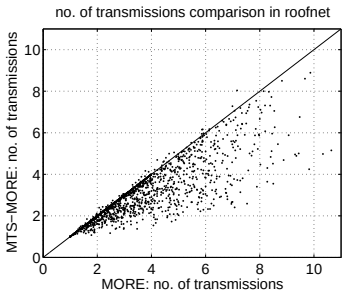

Fig. 8. No. of transmissions

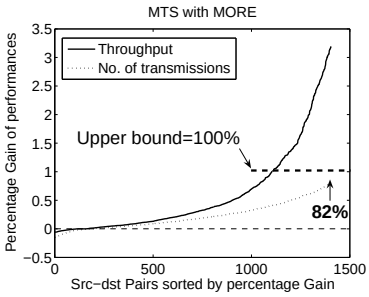

Fig. 9. Percentage gain in MORE

When using MTS (instead of ETX) in MORE to select forwarder lists, we obtain similar results. Figs.7, 8 and 9 show that 93.18% src-dst pairs have fewer number of transmissions, and 91.35% src-dst pairs have larger throughput under MTS-MORE than under ETX-MORE. From Fig.9, we see that MTS can reduce the number of transmissions by up to 82%, and increase the throughput by up to 322% over ETX in MORE. The reason why MTS-MORE achieves better performance than MTS-ExOR is that MORE uses random network coding, instead of the strict scheduling and ACK mechanisms in ExOR, to avoid duplicate transmissions. Hence MORE is more likely to approach the perfect ACK assumption. As a result, MTS-MORE attains better performance gain than MTS-ExOR.

VI. Conclusion

In this paper we established a general theory for analyzing the forwarder list selection problem, and developed an optimal forwarder selection algorithm–the *minimum transmission selection (MTS)* algorithm–which minimizes the expected number of transmissions under the perfect ACK assumption. We showed how this assumption can be relaxed in practice to account for unreliable and asymmetric link qualities. Our theory and algorithm can also be generalized to other routing objectives such as minimizing the expected transmission time or energy consumption in opportunistic routing. Through extensive simulations using the MIT Roofnet dataset, we demonstrated that in more than 90% cases the MTS algorithm outperforms the forwarder selection scheme used in ETX based opportunistic routing protocols such ExOR and MORE, with performance gains up to 82% in terms of the number of transmissions, and up to 322% in terms of throughput.

References

[1] MIT roofnet. *http://pdos.csail.mit.edu/roofnet/*.

[2] S. Banerjee and A. Misra. Minimum energy paths for reliable communication in multi-hop wireless networks. In *Proceedings of Mobihoc*, 2002.

[3] S. Biswas and R. Morris. Exor: Opportunistic routing in multi-hop wireless networks. In *Proceedings of ACM SIGCOMM*, Philadelphia, Pennsylvania, August 2005.

[4] S. Chachulski, M. Jennings, S. Katti, and D. Katabi. Trading structure for randomness in wireless opportunistic routing. In *Proceedings of ACM SIGCOMM*, 2007.

[5] R. R. Choudhury and N. H. Vaidya. Mac-layer anycasting in ad hoc networks. *SIGCOMM Comput. Commun. Rev.*, 34(1):75–80, 2004.

[6] H. Dubois-Ferriere, M.Grossglauser, and M. Vetterli. Least-cost opportunistic routing. In *Allerton Conference on Communication, Control, and Computing*, 2007.

[7] K. Fall and K. Varadhan. *The ns-2 Manual*. The VINT Project, UC Berkeley, LBL, and Xerox PARC, 2003,.

[8] D. B. Johnson and D. A. Maltz. Dynamic source routing in ad hoc wireless networks. In *Mobile Computing*, pages 153–181. Kluwer Academic Publishers, 1996.

[9] S. Katti, D. Katabi, H. Balakrishnan, and M. Medard. Symbol-level network coding for wireless mesh networks. In *Proceedings of ACM SIGCOMM*, Seattle, WA, August 2008.

[10] R. Laufer and L. Kleinrock. Multirate anypath routing in wireless mesh networks. Technical Report UCLA-CSD-TR080025, Computer Science Department, University of California at Los Angeles, 2008.

[11] Y. Li, W. Chen, and Z.-L. Zhang. Design of forwarder list selection scheme in opportunistic routing protocol. Technical Report TR-08-033, CSE Department of University of Minnesota, 2008.

[12] Y. Lin, B. Li, and B. Liang. Codeor: Opportunistic routing in wireless mesh networks with segmented network coding. In *Proceedings of the sixteenth IEEE International Conference on Network Protocols, ICNP*, 2008.

[13] X. Liu, E. K. P. Chong, and N. B. Shroff. A framework for opportunistic scheduling in wireless networks. *Comput. Netw.*, 41(4):451–474, 2003.

[14] C. Perkins and E. Royer. Ad hoc on demand distance vector routing, mobile computing systems and applications. In *Proceedings of WMCSA*, 1999.

[15] X. Zhang and B. Li. Dice: a game theoretic framework for wireless multipath network coding. In *Proceedings of ACM MobiHoc*, 2008.

[16] Z. Zhong and S. Nelakuditi. On the efficacy of opportunistic routing. In *Proceedings of the 4th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks, SECON*, June 2007.