

# Erasure-Coding Based Utility Routing in Multi-hop Wireless Networks

Mingming Lu

School of Information Science and Engineering  
Central South University  
Email: lumingming@mail.csu.edu.cn

Jie Wu

Department of Computer Science and Engineering  
Florida Atlantic University  
Email: jie@cse.fau.edu

**Abstract**—Recently, erasure coding has become widely used to compensate for the low packet delivery ratio of multi-hop wireless networks. Most existing works assume that the number of redundant erasure-coded packets is unlimited for a message. In this paper, we relax this assumption and integrate the packet quantity into a routing problem in order to achieve a good balance between energy cost and reliability (packet delivery ratio). In doing so, we encounter two major challenges in regards to our new routing problem: one is determining the optimal quantity of redundant packets and the other is determining the optimal routing path. We introduce a benefit value to reflect the trade-off between cost and reliability, and design a single metric (expected utility) by integrating benefit value, cost, and reliability. Based on the expected utility metric, we explore the optimality in both path coding and source coding models. In the path coding model, we design an optimal algorithm, and in the source coding model we propose a heuristic solution. Results from extensive simulations on our custom simulator verify our claims.

**Index Terms**—Energy cost, erasure coding, multi-hop wireless networks, reliability, routing, utility.

## I. INTRODUCTION

In multi-hop wireless networks, the unreliable communication [1]–[3] caused by the unstable wireless medium is one of the major challenges. Compared to other methods, such as retransmissions and backup paths, erasure coding has better performance in increasing packet delivery reliability. Erasure coding is a coding technique that can convert a message into a larger set of coded blocks such that any sufficiently large subset of coded blocks can reconstruct the original message.

In this paper, we integrate erasure coding into the routing problem in multi-hop wireless networks. Most existing works [4], [5] that have applied erasure coding to routing problems in unreliable environments assume that there is no limitation on the number of redundant erasure-coded packets for a message. This assumption is not realistic because the overhead of message transmission can be extremely high without proper restrictions on redundant packets. Therefore, we consider an erasure-coding based routing scheme with a pre-defined quantity of redundant packets. The quantity of redundant packets should reflect the importance of corresponding packets because an important packet requires high reliability; and the increment of quantity on redundant packets can directly increase the transmission reliability.

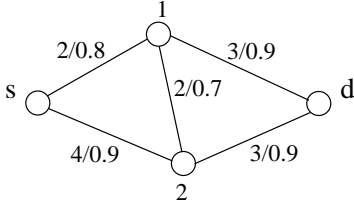
The challenge of integrating erasure coding into a routing scheme lies in the fact that the routing algorithm needs to determine not only the optimal quantity of redundant packets,

but also the optimal routing path. We adopt a market value to reflect the importance of a packet. In a market, an item or a service with higher market value usually corresponds to higher quality (or quality of service), and thus incurs higher costs. In analogy, different types of packets have different market values to the routing source. The source would rather deliver the packet with higher market value through the route with higher reliability; thereby, increasing the quantity of redundant packets at the expense of higher cost. To discriminate the market value of the packet from the general market value in the real market, we adopt the benefit value to denote the former.

The objective of our routing scheme is to determine an optimal combination of a routing path and redundant packets so that the expected utility, which is defined as the expected benefit value (defined as reliability times benefit value) minus transmission cost, can be maximized [6]. For a specific type of packet, the benefit value is fixed. However, the expected benefit value varies because the benefit is realized only when the packet is successfully delivered. The cost and the reliability depend on both the specific routing path and the quantity of redundant packets.

We consider two coding models: *path coding* and *source coding*. Path coding differs from source coding in that path coding allows intermediate nodes to generate erasure-coded packets instead of the source alone in the source coding. In the path coding model, we design an optimal algorithm that can determine the optimal combination of routing path and quantity of redundant packets. In the source coding model, we propose a heuristic solution.

The major contributions of this paper can be enumerated as follows: 1) We integrate erasure coding into a utility-based routing scheme where the maximization of utility depends not only on the routing path but also on the quantity of redundant packets. 2) Through mathematical analysis, we prove the existence of the optimal quantity of redundant packets for both path coding and source coding models. 3) We design an optimal algorithm to determine the optimal combination of the routing path and the quantity of redundant packets in the path coding model. 4) We design distributed implementations for a single source-destination pair and all source-destination pairs. 5) Through experimental studies, we compare the performance of the erasure-coding based utility routing with that of non-erasure coding routing under different settings, and the performance of different forwarding schemes in the path coding model.



	$u_d$	$u_1$	$u_2$	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. 1. An example of a simple network.

## II. PRELIMINARIES

### A. Utility-Based Routing

In [6], a unicast routing problem is considered, where 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 probability of success (reliability) for transmitting a single packet from  $s$  to  $d$  are  $c$  and  $p$ , respectively. If a transmission is successful,  $s$  will obtain benefit  $v$ , consume cost  $c$ , and its utility is  $v - c$ . Otherwise, its utility is  $0 - c$ . Since the probability of success is  $p$ , and the failure probability is  $1 - p$ , the expected utility is as follows:

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

In general, consider a multi-hop path  $R = \langle s = 1, 2, \dots, m-1, d = m \rangle$ . The corresponding expected utility is as follows:

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

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

The REUs of nodes on each path are listed in the table at the bottom of Figure 1, where in each cell, two values separated by “/” represent the REUs under benefit of 20 and 30, respectively. If the benefit  $v = 20$ , the optimal path is  $\langle s, 1, d \rangle$ , and if  $v = 30$ , the optimal route is  $\langle s, 2, d \rangle$ . This reflects the property of the utility-based routing: the optimal route depends not only on the network topology, but also on the benefit value. The search for the path with maximal expected utility can be executed backward, starting from the destination, similar to Dijkstra’s shortest-path algorithm.

### B. Erasure Coding

Erasure coding [4] is a coding technique that converts a message into a set of coded packets such that any sufficiently large subset of the coded packets can be used to reconstruct the original message. In this paper, we assume that the original message has been split into  $k$  equal-sized packets. From the angle of linear algebra, each packet split from the original message can be regarded as a variable, and an erasure-coded packet is a linear combination of the  $k$  original packets. This can be expressed as a linear equation where the left hand side of the equation is the linear combination of the  $k$  original packets, and the right hand side is the erasure-coded packet. As long as  $k$  linearly independent coded packets are given, the original message can be reconstructed by solving the  $k$  linear independent equations associated with the  $k$  coded packets.

In order to increase the reliability of packet delivery in multi-hop wireless networks,  $t$  ( $t > k$ ) erasure-coded packets will be sent by the source. If any  $k$  out of  $t$  packets are received by the destination, the original message can be reconstructed. Obviously, the increment of the value of  $t$  can increase the reliability of packet delivery, but it is at the expense of transmission cost. Hence, a trade-off exists between reliability and cost. The quantity of erasure-coded packets is the key to balancing the trade-off between reliability and cost.

## III. BASIC MODEL

Considering a source-destination pair  $(s, d)$ ,  $s$  sends a message block instead of a single packet to  $d$ .  $d$  will only obtain a benefit if it receives the whole message. Otherwise,  $d$  obtains nothing. We intend to study the routing algorithm that can integrate erasure coding into route discovery in order to determine the most efficient method of delivering the message. We measure the efficiency through expected utility, which is defined as the expected benefit minus the expected cost.

We first consider a simple case, where source  $s$  can directly communicate with destination  $d$ , but packet loss exists. In order to increase the chance that  $d$  will receive the message block,  $s$  breaks the message block into  $k$  (fixed) packets and generates  $t_s - k$  (adjustable) redundant packets through erasure coding. As long as  $d$  can receive at least  $k$  packets from the  $t_s$  packets sent by  $s$ ,  $d$  can restore the original message block through decoding.

According to Formula (2), to calculate the expected utility, we first need to calculate the path reliability/cost. Based on the definition of  $p$  and  $c$  introduced in Section II-A, we can derive the expression for the *path reliability* (denoted as  $P_{sd}(t_s)$ ), which is the probability of success that destination  $d$  will receive sufficient packets ( $k$  packets) to restore the message block from source  $s$ . Since the path reliability is the probability that  $d$  receives at least  $k$  packets, we can first calculate the probability that  $d$  receives exactly  $t$  ( $t \geq k$ ) packets, and then sum up all such probabilities.

Without loss of generality, we assume that the probability of success of each transmission is independent. Therefore, each transmission can be regarded as a Bernoulli experiment; hence, the probability of receiving  $t$  packets follows binomial distribution,  $\binom{t_s}{t} \cdot p^t \cdot (1 - p)^{t_s - t}$ , where  $\binom{t_s}{t}$  is binomial coefficient,

denoting the number of ways that exactly  $t$  out of  $t_s$  packets are received by  $d$ , regardless of their order. By summing up the total  $t_s - k + 1$  probabilities that exactly  $t$  out of  $t_s$  packets are received by  $d$  ( $k \leq t \leq t_s$ ), we can derive the path reliability:

$$P_{sd}(t_s) = \sum_{t=k}^{t_s} \binom{t_s}{t} p^t (1-p)^{t_s-t}. \quad (3)$$

Since  $s$  needs to transmit  $t_s$  packets and each packet consumes cost  $c$ , the corresponding path cost is

$$C_{sd}(t_s) = t_s \cdot c. \quad (4)$$

Hence, the expected utility is

$$U_{sd}(t_s) = P_{sd}(t_s) \cdot v - C_{sd}(t_s). \quad (5)$$

We observe that an optimal value of  $t_s$  exists, with which the expected utility can be maximized. We formalize this property in the following theorem.

*Theorem 1:* For a single link, an optimal quantity of erasure-coded packets exists such that the expected utility through the link can be maximized.

The optimal quantity of erasure-coded packets can be computed using Newton's method [7].

#### IV. THE PATH-CODING MODEL

We study here a general case - a multi-hop path. In the path coding model, we consider two types of forwarding schemes: *conservative forwarding* and *aggressive forwarding*. In the conservative forwarding scheme, intermediate nodes will not forward packets until they receive enough packets to reconstruct the original message. In the aggressive forwarding scheme, intermediate nodes immediately forward each received packet.

A cost-delay trade-off exists between these two schemes. Compared to conservative forwarding, aggressive forwarding has shorter delay, but it is at the expense of larger expected cost. If there is no information loss among intermediate nodes during message transmission (i.e., each intermediate node can reconstruct the original message based on received packets), the energy costs of these two forwarding schemes are the same. Otherwise, aggressive forwarding will have a higher energy cost because intermediate nodes forward erasure-coded packets that cannot be used to reconstruct the original message.

##### A. Conservative Forwarding Scheme

We first consider path reliability, which is a function of the quantity of erasure-coded packets. Without loss of generality, we consider a path  $R = \langle 1, 2, \dots, m \rangle$ , where  $s = 1$  and  $d = m$  in the rest of the paper, and assume that each node  $j \neq d$  sends  $t_j$  ( $t_j \geq k$ ) packets if it receives enough packets from its upstream neighbor  $j - 1$ . We use  $P_{ji}(t_j)$  to denote the reliability of the sub-path from  $j$  to  $i$  along  $R$ . Since no intermediate nodes will send erasure-coded packets until they receive enough packets to effectively reconstruct the original message, path reliability  $P_{ji}(t_j)$  can be decomposed as the product of  $P_{j,j+1}(t_j)$  and  $P_{j+1,i}(t_{j+1})$ . More specifically, path reliability  $P_{jd}$  is:

$$P_{jd}(t_j) = P_{j,j+1}(t_j) \cdot P_{j+1,d}(t_{j+1}), \quad (6)$$

where  $P_{j,j+1}(t_j)$  can be obtained through Formula (3). Hence, by recursively applying Formula (6), we can compute the reliability of any sub-path of  $R$ .

Then, we consider the cost in the conservative forwarding scheme. Considering path  $R$ , the cost of link  $(m-1, d)$  is  $C_{m-1,d}(t_{m-1}) = t_{m-1} \cdot c_{m-1,d}$ . The cost of the sub-path from  $m-2$  to  $d$ , i.e.,  $C_{m-2,d}(t_{m-2})$ , consists of two parts: the cost of link  $(m-2, m-1)$ , i.e.,  $t_{m-2} \cdot c_{m-2,m-1}$ , and the expected cost of link  $(m-1, d)$ . The latter is the product of the reliability of link  $(m-2, m-1)$ ,  $P_{m-2,m-1}(t_{m-2})$ , and the cost of link  $(m-1, d)$ ,  $C_{m-1,d}(t_{m-1})$  since  $m-1$  will not send packets unless it receives at least  $k$  packets. In general, the cost of the sub-path from  $j$  to  $d$  is as follows:

$$C_{j,d}(t_j) = t_j \cdot c_{j,j+1} + P_{j,j+1}(t_j) \cdot C_{j+1,d}(t_{j+1}). \quad (7)$$

Through Formulas (6), (7), and the recursive calculation of Formula (2) in Section II-A, we have:

$$U_{jd}(t_j) = P_{j,j+1}(t_j) \cdot U_{j+1,d}(t_{j+1}) - t_j \cdot c_{j,j+1}. \quad (8)$$

By comparing Formula (8) with Formula (1), we can identify their similarities. Hence, the routing problem from source  $s$  to destination  $d$  can also be decomposed into two sub-problems: the sub-problem from  $s$  to its next-hop 2 and the sub-problem from node 2 to  $d$ . This decomposition is similar to the recursive definition of the shortest path. The major difference is that our solution needs to determine the number of erasure-based packets sent by each intermediate node, and the next-hop as well. Another difference is that our model measures the distance to the destination through expected utility instead of cost.

The calculation of the REU starts from the destination with the initial REU equal to the benefit value (the REUs of all other nodes being  $-\infty$ ). The REU will be reduced at each intermediate node going backwards from the destination to the source. In each iteration, our algorithm not only finds the node that reduces the expected utility to the least, but also identifies the optimal number of erasure-coded packets that should be transmitted by that node.

Our algorithm is presented in Algorithm 1, where  $Q$  represents the set of nodes, the REUs of which have been maximized, and  $\mathcal{N}$  denotes the set of remaining nodes, the REUs of which have not been maximized. Initially,  $\mathcal{N}$  contains all nodes and  $Q$  is empty.  $t_j$  represents the number of erasure-coded packets sent by node  $j$ , and  $t_j^*$  is the corresponding optimal value of  $t_j$ . The existence of the optimal value is guaranteed by Theorem 1. We also use  $U_{ld}^*$  to represent  $U_{ld}(t_l^*)$ . Algorithm 1 is optimal in terms of maximizing the expected utility from  $s$  to  $d$ . The proof of the optimality is similar to that of Dijkstra's shortest path algorithm.

After route discovery, the source breaks its message into  $k$  packets, generates  $t_s^* - k$  redundant packets, and sends these  $t_s^*$  packets to its next hop on the optimal path. After receiving packets from the previous hop, each intermediate node counts the received packets. If the number of received packets  $t < k$ , the intermediate node will not forward the packets. Otherwise, intermediate node  $j$  will forward the received packets. In that case, node  $j$  also needs to check whether the number of received packets is larger than or equal to  $t_j^*$ . If  $k \leq t < t_j^*$ , node  $j$  will generate additional  $t_j^* - t$  erasure-coded packets, and forward these newly-generated

---

**Algorithm 1** MaxREUConservative( $\mathcal{N}, s, d, v$ )

---

- 1: Initialization;
- 2: **while**  $s \notin Q$  **do**
- 3:   Find node  $l \in \mathcal{N}$  with the largest REU;
- 4:   Remove  $l$  from the  $\mathcal{N}$  into  $Q$ ;
- 5:   For each neighbor node  $j \in \mathcal{N}$  of  $l$ , Relax( $l, j$ );

**Relax**( $l, j$ )

- 1: Find  $t_j^*$  to maximize  $P_{jl}(t_j) \cdot U_{ld}^* - C_{jl}(t_j)$ ;
  - 2: Update  $t_j^*$  and  $U_{jd}^*$  if  $U_{jd}^* < P_{jl}(t_j^*) \cdot U_{ld}^* - C_{jl}(t_j^*)$ ;
- 

packets along with the received packets to its next hop. If  $t \geq t_j^*$ , node  $j$  simply forwards any  $t_j^*$  packets from the  $t$  received packets.

To illustrate Algorithm 1, we give an example as shown in Figure 2, where the benefit  $v$  is set to 50 and  $k$  is set to 2. Initially, node  $d$  is selected and removed from the set of nodes  $\mathcal{N}$ . Its two neighbors 1 and 2 are relaxed. After relaxation, the REUs of nodes 1 and 2 are updated from 0 to 36 and 46.6, respectively. The number of erasure-coded packets are 2 and 3 for nodes 1 and 2, respectively. Then, node 2, which is the node with the largest REU among the remaining nodes in  $\mathcal{N}$ , is selected and removed from  $\mathcal{N}$ . Node 2 relaxes its neighbor  $s$  and 1. Only source  $s$ ' REU can be updated from 0 to 22.8 and its quantity of erasure-coded packets is 3. Then node 1 will be selected and removed from  $\mathcal{N}$ . Node 1 will also relax source  $s$ . After relaxation,  $s$ ' REU will be updated from 22.8 to 23 and its quantity of erasure-coded packets will be updated from 3 to 4. Hence, the optimal path is  $\langle s, 1, d \rangle$ , where the quantities of erasure-coded packets of nodes  $s$  and 1 are 4 and 2, respectively. Note that the optimal path with no erasure coding is  $\langle s, 2, d \rangle$ . This verifies our argument in the introduction that separated optimization does not necessarily ensure the optimality of the whole problem since the optimal path without erasure-coding is not necessarily the optimal path with erasure-coding.

### B. Aggressive Forwarding Scheme

The aggressive forwarding scheme can expedite the packet delivery process by allowing intermediate nodes to directly forward each received packet without waiting to check the information loss, which occurs if any intermediate node eventually receives less than  $k$  packets. Although intermediate nodes in the aggressive forwarding scheme will not generate new erasure-coded packets unless they receive at least  $k$  packets, an extra cost is still introduced in the case of information loss, compared with the conservative forwarding scheme. For example, for a path with only two links:  $(s, 2)$  and  $(2, d)$ , the extra cost is introduced at node 2 in the event that less than  $k$  packets from  $s$  are received by node 2. If node 2 receives exactly  $t$  ( $t < k$ ) packets, the extra cost is  $t \cdot c_{2d}$  since node 2 will not forward these packets in the conservative forwarding scheme. There are a total of  $k - 1$  in such cases. By summing them up, the total expected extra cost is:

$$C'_{2d}(t_s^*) = \sum_{t=1}^{k-1} \binom{t_s^*}{t} p_{s2}^t (1 - p_{s2})^{t_s^* - t} \cdot t \cdot c_{2d}.$$

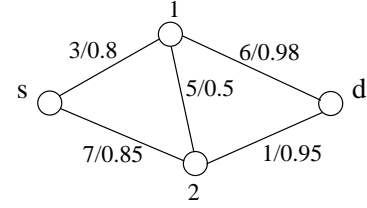


Fig. 2. The illustration of the conservative forwarding scheme.

For a general path  $R$  with  $m$  nodes, where  $s = 1$  and  $d = m$ , if node  $l$  ( $1 < l < m$ ) is the first *failure* node (receiving less than  $k$  packets), the extra costs will occur on all *downstream intermediate nodes* of node  $l$ , i.e., any node  $j$  ( $l < j < m$ ). Note that the extra costs are all expected values. By enumerating all first-failure nodes, we can enumerate all possible cases of failure. Supposing that  $t_{j-1}$  ( $t_{j-1} < k$ ) is the number of packets transmitted by node  $j$ 's upstream neighbor  $j - 1$ , the sum of the expected extra costs of node  $j$  and all its downstream nodes can be expressed as:

$$C'_{jd}(t_{j-1}) = \sum_{t=1}^{t_{j-1}} \binom{t_{j-1}}{t} p^t (1-p)^{t_{j-1}-t} (tc + C'_{j+1,d}(t)), \quad (9)$$

where  $p_{j-1,j}$ , and  $c_{j,j+1}$  are abbreviated as  $p$  and  $c$ . We assume that  $C'_{dd}(t) = 0$  because  $d$  will not forward packets. By recursively applying Formula (9), we can derive the expression for the total expected extra cost that occurs in the event that node  $l$  is the first failure node:

$$C'_{ld}(t_{l-1}) = \sum_{t=1}^{k-1} \binom{t_{l-1}}{t} p^t (1-p)^{t_{l-1}-t} (t \cdot c + C'_{l+1,d}(t)), \quad (10)$$

where  $t_{l-1} \geq k$ , and  $p_{l-1,l}$  and  $c_{l,l+1}$  are abbreviated as  $p$  and  $c$ , respectively. Since any two different intermediate nodes cannot be the first failure node at the same time, the total extra cost is simply the sum of all  $C'_{ld}(t_{l-1})$ , where  $1 < l < m$ .

We observe that the precondition of node  $l$  being the first failure node is the same as the precondition that node  $l-1$  spends energy on transmitting packets in the conservative forwarding scheme, i.e., all upstream nodes of  $l$  have no information loss. Hence, we can integrate the extra cost  $C'_{ld}(t_{l-1})$  into the cost of node  $l-1$ , i.e., the cost of link  $(l-1, l)$ . By doing so, we can still apply a Dijkstra-based algorithm.

More specifically, in the Relax procedure of Algorithm 1,  $C_{jl}(t_j)$  is equal to  $t_j \cdot c_{jl}$  in the conservative forwarding scheme, while, in the aggressive forwarding scheme, it is  $C_{jl}(t_j) = t_j \cdot c_{jl} + C'_{ld}(t_j)$ , where  $C'_{ld}(t_j)$  is calculated based on Formula (10), in which  $l-1$  is replaced by  $j$ , if  $l \neq d$ . If  $l = d$ ,  $C'_{ld}(t) = 0$  for any  $t \geq 0$ . Moreover, after relaxation, node  $j$  needs to calculate and record  $C'_{ld}(t)$  ( $0 < t < k$ ) based on Formula (9), where  $j$  is replaced by  $l$ . The formal description of the algorithm is presented in Algorithm 2. Algorithm 2 is optimal in terms of maximizing the expected utility in the aggressive forwarding scheme. Hence, we have the following theorem.

**Theorem 2:** Algorithm 2 is optimal for the aggressive forwarding scheme.

---

**Algorithm 2** MaxREUAggressive( $\mathcal{N}, s, d, v$ )

---

1: The same as Algorithm 1

**Relax**( $l, j$ )

```
1:  $C_{jl}(t_j) \leftarrow t_j \cdot c_{jl} + C'_{ld}(t_j)$ ;  
2: Find  $t_j^*$  to maximize  $P_{jl}(t_j) \cdot U_{ld}^* - C_{jl}(t_j)$ ;  
3: if  $U_{jd}^* < P_{jl}(t_j^*) \cdot U_{ld}^* - C_{jl}(t_j^*)$  then  
4:   Update  $t_j^*$  and  $U_{jd}^*$ ;  
5:   for  $0 < t < k$  do  
6:     if  $l = d$  then  
7:        $C'_{ld}(t) \leftarrow 0$ ;  
8:     else  
9:        $C'_{ld}(t) \leftarrow \sum_{i=1}^t \binom{t}{i} p_{j,i}^i (1 - p_{j,i})^{t-i} (i c_{l,l+1} + C'_{l+1,d}(i))$ ;
```

---

Compared with the conservative forwarding scheme, the aggressive forwarding scheme has a shorter delay. Now, we will formally analyze their delay difference. If the data rate is  $r$ , one packet will be sent out every  $1/r$  seconds. In this scheme, if there is no packet loss, the destination will receive one packet every  $1/r$  seconds after the arrival of the first packet. In the conservative forwarding scheme, an additional  $(k-1) \cdot 1/r$  delay is introduced at each intermediate node because each intermediate node will not forward the first  $k-1$  packets received until it receives the  $k$ -th packet. Hence, the aggressive forwarding scheme saves  $m \cdot (k-1) \cdot 1/r$  seconds through a path with  $m$  intermediate nodes in the event of no packet loss. In the event of packet loss, each event of packet loss incurs an additional  $1/r$  second delay, but it is the same for both forwarding schemes. Hence, the saved delay is also  $m \cdot (k-1) \cdot 1/r$  in the event of packet loss. Note that we count the saved delay only when the destination receives at least  $k$  packets, the probability of which is the same for both forwarding schemes. The above delay analysis depends on the assumption that intermediate nodes will not generate new erasure-coded packets until they receive at least  $k$  packets. This assumption is used to ensure that any  $k$  erasure-coded packets out of the packets sent by any intermediate node are linearly independent.

### C. Distributed Implementation

In the case of a single source-destination pair, we develop a distributed implementation that computes REU in a distributed manner. REU could be treated as the summary of local link state information. Each node needs not propagate all available local link state information to its neighbors. Instead, it propagates summarized routing information. In the conservative forwarding scheme, the summarized routing information is simply the REU to the destination, while in the aggressive forwarding scheme, it further includes the cost to the next-hop node, and the extra cost calculated based on Formula (9).

The distributed implementation can be gracefully integrated in a reactive routing protocol, such as AODV [8] or DSR [9], where two phases are used. In the route discovery phase, the source broadcasts a RREQ (route request) to its neighbors. The RREQ is propagated in the network until it gets to the destination, which then initiates a RREP (route reply) containing relevant information following the reverse link leading to the source.

- 1) The source sends out a message to inform the destination of its benefit.
- 2) The destination broadcasts its REU to initialize a route discovery phase that will form a global directed flooding tree rooted at the destination.
- 3) Upon receiving the first REU, each node  $i$ , including the source, calculates its initial REU (i.e.,  $u_i$ ) and sets a timer  $w_i$ , which is proportional to  $v - u_i$  and time period, since the original message was sent out at the destination.
- 4) Before timeout, each node improves its REU based on the received REUs of its neighbors and adjusts its timer.
- 5) After timeout, each intermediate node computes and sends out its summarized routing information to all neighbors.

The initial value of the timer is a function in which  $w_j$  is proportional to  $v - u_i$  and is adjusted by  $x^* - x$  (time period), where  $x^*$  is the current time and  $x$  is the time the message was sent out at the destination. We also assume the initial time  $x$  is attached when the destination initiates the priority discovery phase. The higher the  $u_i$  is, the shorter the amount of time that node  $i$  will back-off before it sends out its priority. Whenever a node  $j$  receives messages from its neighbors that improve its REU, it will reduce the remaining back-off time accordingly. Once the timer is properly set, the node with the maximum REU among all back-off nodes will send out its reply first, which includes its REU.

Although the distributed implementation is an approximation due to transmission delay, 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 REU that summarizes the link state information will be propagated.

We use Figure 2 to illustrate the distributed implementation. After the route request phase,  $d$  broadcasts RREP(50) immediately. Assume path  $\langle d, 1, s \rangle$ 's traffic is heavy, so  $(d, 1)$  delays  $1ms$  (milliseconds) and  $(1, s)$  delays  $4ms$ . But path  $\langle d, 2, s \rangle$ 's traffic is light, where  $(d, 2)$ 's delay is  $0.3ms$  and  $(2, s)$ 's delay is  $0.2ms$ . The delay between  $(1, 2)$  is  $0.1ms$ . Assume that the time  $d$  broadcasts RREP(50) is  $0ms$ . At time  $0.3ms$ , node 2 receives the RREP(50), updates its REU to 46.6, and backs off  $0.34 - 0.3 = 0.04ms$ . At time  $0.34ms$ , it will broadcast RREP(46.6) to its neighbors. At time  $0.44ms$ , node 1 receives RREP(46.6), updates its REU to 1.65, and backs off  $4.84 - 0.44 = 4.4ms$ . At time  $0.54ms$ ,  $s$  receives RREP(46.6), updates its REU to 22.8, and backs off  $3.72 - 0.54 = 3.18ms$ . At time  $1ms$ , node 1 receives RREP(50), updates its REU to 36, and backs off  $1.4 - 1 = 0.4ms$ . At time  $1.4ms$ , node 1 broadcasts RREP(36) to its neighbors. At time  $3.72ms$ ,  $s$  ends its back-off time and finds the path  $\langle s, 2, d \rangle$ . In this example, the potential optimal path is missed due to transmission delay.

## V. EXTENSIONS

### A. All Source-Destination Pairs

We adopt a modified distributed Bellman-Ford algorithm for the case of all source-destination pairs. In the traditional distributed Bellman-Ford algorithm [10], each node exchanges cost and

routing information with its neighbors on an interactive basis until the routing table at each node converges to the appropriate shortest path entries. In our modified version, nodes also exchange reliability information. Each node computes the maximum REU path to every other node based on exchanged routing information. For a specific benefit value, each entry of our routing table is (*destination, next-hop, packet quantity*) instead of (*destination, next-hop*).

Once a node finds a new path (to a destination) that has a larger REU than the current path, it will update the current path with the new path by modifying the corresponding next-hop and quantity of erasure-coded packets. After updating, the node will broadcast this new path to its neighbors. When its neighbors receive the updated routing information, they will check whether the new path has a larger REU than their current paths. If so, they repeat the same update and broadcast process. Otherwise, they simply discard it.

The distributed implementation consists of two parts: an initialization step and a maximum-REU calculation step that is repeated until the algorithm has been completed. Here, the maximum REU represents the one between a given node and the destination node for a specific benefit value. The distributed implementation ends with all nodes labeled with their maximum REUs, the corresponding next-hop nodes, and quantity of erasure-coded packets to each destination node. For each destination, each node  $j$  has the label  $(l, U_j, t_j)$  where  $l$  is the next-hop node along with the current maximum-REU path,  $U_j$  represents the current value of the REU from  $j$  to the destination, and  $t_j$  is the quantity of erasure-coded packets sent by node  $j$ .

- Initialization: With node  $d$  being the destination node, set  $U_d = v$  and label all other nodes  $(\cdot, -\infty, 0)$ .
- Labeling of all nodes: For each node  $j \neq d$  does the following: Upon receiving new routing information from neighbor  $l$ ,  $j$  will update  $(i, U_j, t_j)$  if the REU through  $l$  is larger than the REU through  $i$ .  $j$  broadcasts its updated routing information to its neighbors thereafter.

The routing information is different between conservative and aggressive forwarding schemes. In the conservative forwarding scheme, a node only needs to broadcast its updated REU to its neighbors, while in the aggressive forwarding scheme, a node  $l$  further needs to broadcast the cost  $c_{l,l+1}$ , and  $C'_{l+1,d}(t)$  for all  $t$  satisfying  $0 < t < k$ .

For the update, node  $j$  calculates the optimal value of  $t_j$  (denoted as  $t_j^*$ ) that maximizes  $P_{j,l}(t_j) \cdot U_l - C_{j,l}(t_j)$ . If  $U_j$  is less than  $P_{j,l}(t_j^*) \cdot U_l - C_{j,l}(t_j^*)$ ,  $j$  sets  $U_j = P_{j,l}(t_j^*) \cdot U_l - C_{j,l}(t_j^*)$ , and  $i = l$ . In the conservative forwarding,  $C_{j,l}(t_j) = t_j \cdot c_{j,l}$ , while, in the aggressive forwarding,  $C_{j,l}(t_j) = t_j \cdot c_{j,l} + C'_{l,d}(t_j)$ , where  $C'_{l,d}(t_j)$  is obtained through Formula (10).

### B. Source Coding Model

The source coding model is different from the path coding model in that only the source itself can generate erasure-coded packets in the source coding model. In the source coding model, we consider only the conservative forwarding scheme because the aggressive forwarding scheme can be solved based on the

solution to the conservative forwarding scheme as in the path coding model.

We first consider the path reliability and still use path  $R$  as an example. Unlike the path coding model, in the source coding model, path reliability  $P_{sd}(t_s)$  cannot be simply decomposed into reliability of the link from  $s$  to its next-hop node 2 and reliability of the sub-path from node 2 to  $d$  because the sub-path reliability depends on the exact number of packets received by node 2. For example, if node 2 receives  $t$  packets, the number of packets sent by node 2 must be less than  $t$  since only the source can generate erasure-coded packets. If node 2 receives less than  $k$  packets, node 2 and its downstream nodes (including  $d$ ) cannot reconstruct the original message. The sub-path reliability is 0 in that case. Hence, we only count the case  $t \geq k$ . For each value of  $t$ , suppose that the sub-path reliability is  $P_{2d}(t)$ . Supposing that  $p_{s2}$  is the probability of success of transmitting a single packet from  $s$  to node 2, the probability of exactly  $t$  packets (out of  $t_s$  packets) received by node 2 is  $\binom{t_s}{t} p_{s2}^t (1 - p_{s2})^{t_s - t}$ . By summing up the total  $t_s - k + 1$  cases ( $t$  counted from  $k$  to  $t_s$ ), we obtain the following recursive form for  $P_{sd}(t_s)$ :

$$P_{sd}(t_s) = \sum_{t=k}^{t_s} \binom{t_s}{t} p_{s2}^t (1 - p_{s2})^{t_s - t} P_{2d}(t).$$

In general, for any node  $j$ , the reliability of the sub-path (from  $j$  to  $d$  with  $t_j$  packets forwarded by node  $j$ ) can be expressed as:

$$P_{jd}(t_j) = \sum_{t=k}^{t_j} \binom{t_j}{t} p_{j,j+1}^t (1 - p_{j,j+1})^{t_j - t} P_{j+1,d}(t), \quad (11)$$

where  $j + 1$  is the next-hop node of  $j$  on path  $R$ .

Then, we consider the cost of path  $R$ , the cost of link  $(m - 1, d)$  is  $C_{m-1,d}(t_{m-1}) = t_{m-1} \cdot c_{m-1,d}$ , if node  $m - 1$  forwards  $t_{m-1} (\geq k)$  packets. Similarly, if node  $m - 2$  forwards  $t_{m-2} (\geq k)$  packets, the cost of the sub-path from  $m - 2$  to  $d$  is  $C_{m-2,d}(t_{m-2}) = t_{m-2} \cdot c_{m-2,m-1} + \sum_{t=k}^{t_{m-2}} \binom{t_{m-2}}{t} p_{m-2,m-1}^t \cdot (1 - p_{m-2,m-1})^{t_{m-2} - t} C_{m-1,d}(t)$ . In general, the cost of the sub-path from  $j$  to  $d$  is in the following recursive form:

$$C_{jd}(t_j) = t_j \cdot c + \sum_{t=k}^{t_j} \binom{t_j}{t} p^t (1 - p)^{t_j - t} C_{j+1,d}(t), \quad (12)$$

where  $c_{j,j+1}$  and  $p_{j,j+1}$  are abbreviated as  $c$  and  $p$ , respectively. The expected utility of the sub-path from  $j$  to  $d$  with  $t_j$  packets forwarded by  $j$  should be  $U_{jd}(t_j) = P_{jd}(t_j) \cdot v - C_{jd}(t_j)$ . Combining Formula (11) and Formula (12), we obtain the following recursive form of the expected utility

$$U_{jd}(t_j) = \sum_{t=k}^{t_j} \binom{t_j}{t} p^t (1 - p)^{t_j - t} U_{j+1,d}(t) - t_j \cdot c, \quad (13)$$

where  $t_j \geq k$ , and  $p$  and  $c$  are the abbreviation of  $p_{j,j+1}$  and  $c_{j,j+1}$ , respectively. For  $t_j < k$ , we define  $U_{jd}(t_j) = 0$ . Similar to Theorem 1, we can also prove that an optimal value of  $t_s$  exists in the source coding model.

*Theorem 3:* In the source coding model, an optimal quantity of erasure-coded packets generated by the source exists so that the expected utility through a given path can be maximized.

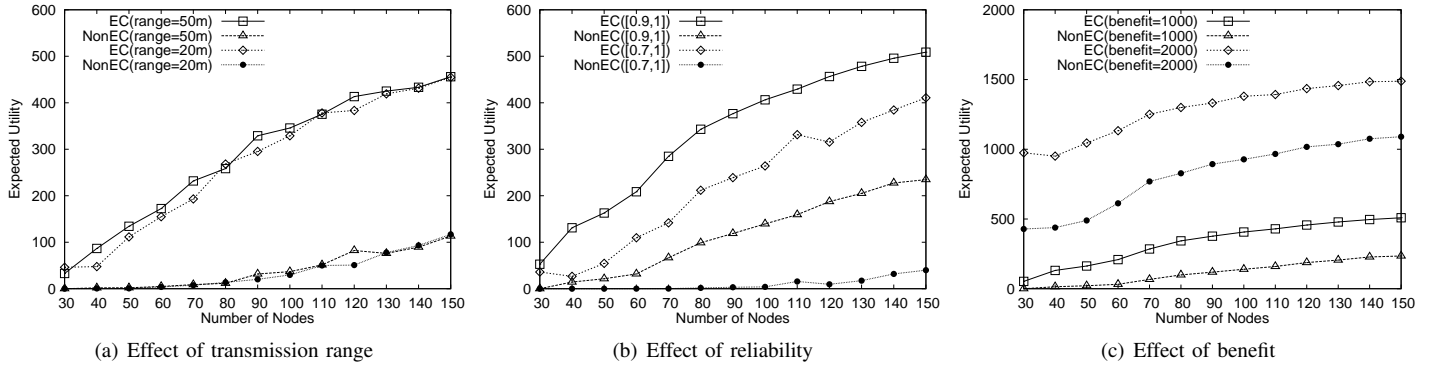


Fig. 3. The effect of (a) transmission range, (b) reliability, and (c) benefit value on both erasure-coding based routing and retransmission-based routing.

Theorem 3 ensures that the optimal quantity of erasure-coded packets exists for any given path. A straightforward solution to the source coding model is to first compute the optimal path without erasure coding, and then determine the optimal quantity of erasure-coded packets. This solution is not necessarily an optimal solution of the source coding model because the separated optimization is not necessarily the optimization of the combined problem. It is still an open problem regardless of whether an efficient solution exists other than enumerating all possible paths and calculating the optimal quantities for all paths.

## VI. RELATED WORK

The existence of low quality links incurs a low packet delivery ratio, and hence reduces the performance of multi-hop wireless networks [1], [11], [12]. A common fault-tolerant technique is redundancy, such as retransmissions or erasure coding. Banerjee and Misra [13] modeled the link cost as a function of energy consumption for a single transmission attempt across the link and the link error rate, and proposed several retransmission aware routing schemes. Banerjee et al. [14] further extended their work by relaxing the assumption of the perfect reliability (zero error rate) in the link layer. Li et al. [15] integrated the power control techniques into the routing problem proposed in [14]. In [6], we further relaxed the assumption of perfect end-to-end reliability, and adopted the expected utility to balance the trade-off between maximization of the packet delivery ratio and minimization of the effective energy consumption.

It has been verified that erasure coding is better than retransmissions in terms of cost and delay [4], [16], [17]. Erasure coding can be used for both inter-flow routing [18] and intra-flow routing [19]. The network coding methods use replication (redundancy) and send identical copies of a message simultaneously over multiple paths to mitigate the effects of a single path failure [20]. Jain et al. [21] and Wang et al. [22] applied erasure coding to delay tolerant networks (DTNs). Cui et al. [23] proposed a jointly opportunistic source coding and opportunistic routing (OSCOR) protocol for correlated data gathering in wireless sensor networks. Cristescu and Beferull-Lozano [24] considered a sensor network measuring correlated data, where the task is to gather all data from the network nodes to a sink.

## VII. SIMULATION

We compare routing schemes with and without erasure coding. For comparison purposes, we calculate the redundancy ratio in our erasure-coding based utility routing scheme, and use this ratio as the average number of retransmissions for each packet at each node (excluding the destination) in the routing scheme without erasure coding. We set up the simulation in a  $100m \times 100m$  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  $(15m, 50m)$  and  $(85m, 50m)$ , 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 a function of the received power at node  $j$  (denoted as  $R_j$ ). In the simulation, we consider  $n$ , the number of nodes as a tunable parameter. We vary the number of nodes from 30 to 150. We also vary the maximum transmission range. The connectivity of the network is controlled solely by the node quantity and maximum transmission range. We use a homogeneous maximum transmission range for simplicity, and evaluate two maximum transmission ranges:  $20m$  and  $50m$ . We also evaluate two benefit values: 1,000 and 2,000. The default benefit value is 1,000.

We first evaluate the impact of the network connectivity on utility-based routing schemes with and without erasure coding. The experimental result is shown in Figure 3(a), where EC and NonEC are used as abbreviations for utility-based routing with and without erasure coding, respectively. The EC scheme has better performance than that of NonEC. The reason is that erasure coding provides redundancy for a whole message and not for a single packet, and thus reduces the redundancy ratio. We also observe that increment of the network connectivity can improve the expected utility obtained through the optimal paths under both schemes. The reason is that the increment of the network connectivity provides more available paths. Since the maximum transmission range has little impact on the increment of expected utility compared with the node density, we set the maximum transmission range to  $20m$  in the rest of the simulations.

We also evaluate the impact of reliability on both schemes. We evaluate two reliability ranges  $[0.7, 1]$  and  $[0.9, 1]$ . The experi-

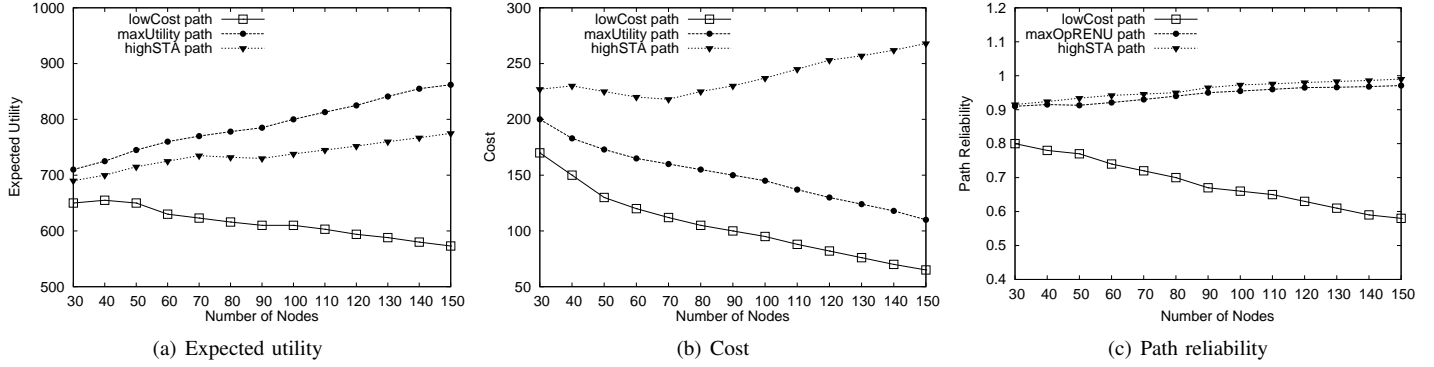


Fig. 4. Performance comparison of three optimal routing paths under three different metrics.

mental results are shown in Figure 3(b). From our results, we can conclude that the reliability range has a significant impact on the expected utility. This is because reliability has a direct impact on the quantity of erasure-coded packets and the route selection. We also observe that the EC scheme has better performance than that of the NonEC scheme under various reliability ranges. We then evaluate the impact of the benefit value on both schemes. We use two benefit values: 1,000 and 2,000. From the simulation result shown in Figure 3(c), we can conclude that the benefit value can increase the expected utility for both routing schemes because the benefit value is strongly correlated to the expected utility according to Formula (2). From Figure 3, we can conclude that the EC scheme is better than the NonEC scheme in terms of maximizing the expected utility.

To illustrate that the utility can be used to efficiently allocate energy cost over networks, we compare it with two other metrics, cost and reliability. For these two metrics, we design the corresponding erasure-coding based cost/reliability routing schemes as follows: 1) we first calculate the redundancy ratio in the erasure-coding based utility routing scheme; 2) we use this ratio as the average redundancy ratio of the erasure-coded packets for each node in the network; 3) we compute the corresponding lowest cost path and the highest reliability path based on the above determined ratio. The simulation results are illustrated in Figure 4, where the lowCost path, maxUtility path, and highSTA path represent the lowest cost path, the maximum expected utility path, and the highest reliability path in the erasure-coding based routing framework, respectively.

Figure 4 compares the three optimal paths under three metrics: expected utility, path cost, and path reliability. Figure 4(a) shows that maxUtility path has the best performance in terms of expected utility. From Figures 4(b) and 4(c), we can see that maxUtility path's performance is second best in terms of path cost and reliability, respectively. The results show that our expected utility metric is useful for evaluating routing performance in multi-hop wireless networks because it achieves a good trade-off between cost and reliability.

In Figure 4(a), in terms of expected utility, highSTA path has better performance than lowCost path because reliability has a greater effect on expected utility than on cost. In Formula (2), if

benefit  $v$  is large enough, the expected utility will decrease by half with the reliability  $P_R$  decreasing by half. However, the expected utility will not decrease too much when the cost  $C_R$  is doubled. Figure 4(a) also shows that the expected utilities of maxUtility path and highSTA path increase as the number of nodes increases, but it decreases for lowCost path. With more nodes, more paths are available. As the number of nodes increases, the selected lowCost path might have more hop counts and hence have lower reliability. The impact of cost decrement cannot make up for the impact of decrement in reliability.

In Figure 4(b), both maxUtility path and lowCost path decrease as the number of nodes increases, but highSTA path does not. As we have argued above, with the increment of node number, the available paths increase, and hence more lower-cost paths will be available. However, the cost of highSTA path does not necessarily decrease. The expected utility metric adequately balances the trade-off between reliability and cost. In Figure 4(c), except for lowCost path, the path reliability of the paths increases with the increasing number of nodes. We have discussed the reason for this in the above arguments. The desirable result is that maxUtility path shows good path reliability. LowCost path is the worst of the three paths in terms of reliability. The result is not surprising because the lowest-cost path usually has more hops in multi-hop wireless networks. The experimental results shown in Figure 4 illustrate that expected utility is an efficient metric to assess the utilization of network resource.

We also compare the performance of the conservative forwarding scheme and the aggressive forwarding scheme through simulation studies. We set  $v = 1,000$ ,  $\alpha = 0.9$ ,  $\beta = 1$ , the maximum transmission range to  $20m$ , and  $k = 5$ . The simulation results are presented in Figure 5. We first compare the expected utilities of both schemes, the result of which is shown in Figure 5(a). It is not surprising that the expected utilities of the conservative forwarding scheme is higher than that of the aggressive forwarding scheme since the latter consumes more energy cost in order to reduce delay. From Figure 5(b), we observe that the conservative forwarding scheme has a higher redundancy ratio when the number of nodes is less than 100. The redundancy ratio of the aggressive ratio increases faster and is almost the same as that of the conservative scheme when the number of nodes is



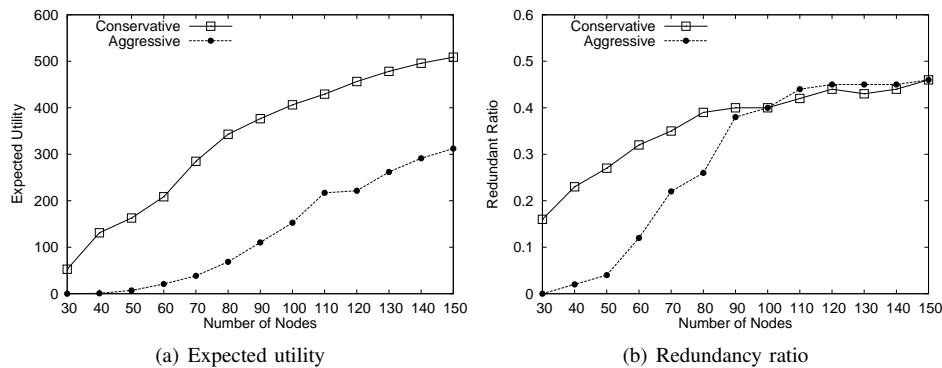


Fig. 5. Performance comparison of the conservative forwarding scheme and the aggressive forwarding scheme.

larger than 100. The reason is that the cost of each link in the conservative forwarding scheme is less than that in the aggressive forwarding scheme. When fewer nodes exist, the average node-to-node distance is relatively long, and so the average transmission cost will be higher. Hence, the aggressive forwarding scheme tends to use fewer redundant packets. When more nodes are available, the average transmission cost decreases quadratically. The impact of the extra cost can be compensated by selecting paths with a shorter average hop-by-hop distance.

The simulation results can be summarized as: 1) The EC scheme has better performance than the NonEC scheme. 2) The EC scheme is sensitive to the range of reliability, the number of nodes, and the transmission range. 3) The expected utility metric based on our erasure-coding based routing scheme has better performance than two other metrics (lowest cost and highest reliability) because it adequately balances the trade-off between reliability and cost. 4) The conservative forwarding scheme saves more energy than the aggressive forwarding scheme.

## VIII. CONCLUSION

In this paper, we relax the assumption of unlimited redundant erasure-coded packets for a message and integrate the quantity of the redundant packets into the routing problem so that we can achieve a good balance between the energy cost and reliability. We introduce a benefit value to reflect the trade-off between cost and reliability, and design a single metric expected utility by integrating benefit value, cost, and reliability. Based on the expected utility metric, we explore the optimality in both path coding and source coding models. In the path coding model, we design an optimal algorithm, and in the source coding model, we propose a heuristic solution, which is verified through experimental study. In the future, we will extend our study to multi-path routing, multi-cast, and opportunistic routing schemes. We will also extend our utility metric to include delay and other parameters, such as data rate.

## ACKNOWLEDGEMENT

This work was supported in part by NSF grants CNS 0422762, CNS 0434533, CNS 0531410, and CNS 0626240.

## REFERENCES

- [1] A. Cerpa, J. L. Wong, M. Potkonjak, and D. Estrin, "Temporal properties of low power wireless links: modeling and implications on multi-hop routing," in *Proceedings of ACM MobiHoc'05*, 2005, pp. 414–425.
- [2] D. Couto, D. Aguayo, B. Chambers, and R. Morris, "Performance of multi-hop wireless networks: shortest path is not enough," *SIGCOMM Computer Communication Review*, vol. 33, no. 1, pp. 83–88, 2003.
- [3] D. A. Maltz, J. Broch, and D. B. Johnson, "Quantitative lessons from a full-scale multi-hop wireless ad hoc network testbed," in *Proceedings of the IEEE WCNC*, 2000.
- [4] M. Ghaderi, D. Towsley, and J. Kurose, "Reliability benefit of network coding," Computer Science Department, University of Massachusetts Amherst, Tech. Rep., February 2007.
- [5] D. S. Lun, M. Medard, and R. Koetter, "Network coding for efficient wireless unicast," *2006 International Zurich Seminar on Communications*, pp. 74–77, 2006.
- [6] M. Lu and J. Wu, "Social welfare based routing in ad hoc networks," in *Proceedings of IEEE ICPP'06*, 2006, pp. 211–218.
- [7] C. T. Kelley, *Solving Nonlinear Equations with Newton's Method*. SIAM, 2003.
- [8] C. Perkins, Ad Hoc On Demand Distance Vector (AODV) Routing IETF, Internet Draft, draft-ietf-manet-aodv-00.txt, November 1997. [Online]. Available: citeseer.ist.psu.edu/article/perkins99ad.html
- [9] D. B. Johnson and D. A. Maltz, *Dynamic Source Routing in Ad hoc Wireless Networks*. Mobile Computing, 1994.
- [10] A. Leon-Garcia and I. Widjaja, *Communication Networks, Fundamental Concepts and Key Architectures*. McGraw-Hill Higher Education, Singapore, International Editions 2000.
- [11] A. Woo, T. Tong, and D. Culler, "Taming the underlying challenges of reliable multihop routing in sensor networks," in *Proceedings of ACM SENSYS'03*, 2003, pp. 14–27.
- [12] J. Zhao and R. Govindan, "Understanding packet delivery performance in dense wireless sensor networks," in *Proceedings of ACM SenSys'03*, 2003, pp. 1–13.
- [13] S. Banerjee and A. Misra, "Minimum energy paths for reliable communication in multi-hop wireless networks," in *Proceedings of ACM MobiHoc'02*, 2002, pp. 146–156.
- [14] Q. Dong, S. Banerjee, M. Adler, and A. Misra, "Minimum energy reliable paths using unreliable wireless links," in *Proceedings of ACM MobiHoc'05*, 2005, pp. 449–459.
- [15] X. Li, Y. Shu, H. Chen, and X. Chu, "Energy efficient routing with unreliable links in wireless networks," in *Proceedings of IEEE MASS'06*, 2006, pp. 160–169.
- [16] M. Kurth, A. Zubow, and J. Redlich, "Cooperative opportunistic routing using transmit diversity in wireless mesh networks," in *Proceedings of IEEE Infocom'08*, 2008.
- [17] K. Zeng, W. Lou, and H. Zhai, "On end-to-end throughput of opportunistic routing in multirate and multihop wireless networks," in *Proceedings of IEEE Infocom'08*, 2008.
- [18] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Médard, and J. Crowcroft, "Xors in the air: practical wireless network coding," in *Proceedings of ACM SIGCOMM'06*, 2006.

- [19] S. Chachulski, M. Jennings, S. K., and D. Katabi, "Trading structure for randomness in wireless opportunistic routing," in *Proceedings of ACM SIGCOMM'07*, 2007.
- [20] A. Vahdat and D. Becker, "Epidemic routing for partially-connected ad hoc networks," CS-2000-06, Duke University, Tech. Rep., 2000.
- [21] S. Jain, M. Demmer, R. Patra, and K. Fall, "Using redundancy to cope with failures in a delay tolerant network," *SIGCOMM Computer Communication Review*, vol. 35, no. 4, pp. 109–120, 2005.
- [22] Y. Wang, S. Jain, M. Martonosi, and K. Fall, "Erasure-coding based routing for opportunistic networks," in *Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking (WDTN'05)*, 2005, pp. 229–236.
- [23] T. Cui, L. Chen, T. Ho, and S. Low, "Opportunistic source coding for data gathering in wireless sensor networks," California Institute of Technology, Tech. Rep., 2007.
- [24] R. Cristescu and B. Beferull-Lozano, "Lossy network correlated data gathering with high-resolution coding," *IEEE/ACM Transactions on Networking*, vol. 14, no. SI, pp. 2817–2824, 2006.

## APPENDIX

**Proof of Theorem 1** We use  $U(t)$ ,  $P(t)$ , and  $C(t)$  to represent the utility, reliability, and cost with  $t$  packets, respectively. We define  $\Delta U(t) = U(t+1) - U(t)$  expressed as follows.

$$\text{Lemma 1: } \Delta U(t) = \binom{t}{k-1} p^k (1-p)^{t+1-k} \cdot v - c.$$

*Proof:* According to Equations (3), (4), and (5), we have

$$U(t) = \sum_{i=k}^t \binom{t}{i} p^i (1-p)^{t-i} \cdot v - t \cdot c. \quad (14)$$

Based on Formula (14), we have  $U(t+1) = \sum_{i=k}^{t+1} \binom{t+1}{i} p^i (1-p)^{t+1-i} \cdot v - (t+1) \cdot c$ , where  $\sum_{i=k}^{t+1} \binom{t+1}{i} p^i (1-p)^{t+1-i}$  can be rewritten as

$$\sum_{i=k}^t \left( \binom{t}{i-1} + \binom{t}{i} \right) p^i (1-p)^{t+1-i} + p^{t+1}. \quad (15)$$

Since  $\binom{t}{i} p^i (1-p)^{t+1-i} + \binom{t}{i} p^{i+1} (1-p)^{t-i} = \binom{t}{i} p^i (1-p)^{t-i}$ , Formula (15) is equal to  $\binom{t}{k-1} p^k (1-p)^{t+1-k} + \sum_{i=k}^t \binom{t}{i} p^i (1-p)^{t-i}$ . Therefore, we have  $U(t+1) = \binom{t}{k-1} p^k (1-p)^{t+1-k} + U(t) - c$ . This lemma is then proved. ■

*Proof:* Based on Lemma 1, to prove the existence of the optimal  $t$  is equal to proving that a  $t^*$  exists so that  $\Delta U(t^* - 1) > 0$  and  $\Delta U(t^*) \leq 0$ .

$$\begin{aligned} \Delta U(t) &= \binom{t-1}{k-1} p^k (1-p)^{t+1-k} \cdot v - c \\ &= \frac{t \cdots (t-k+2)}{(k-1)!} p^k (1-p)^{t+1-k} \cdot v - c \\ &\leq \frac{(k-1) \cdot t}{(k-1)!} p^k (1-p)^{t+1-k} \cdot v - c \\ &= t \cdot (1-p)^t \cdot \frac{p^k \cdot v}{(k-2)!(1-p)^{k-1}} - c \end{aligned}$$

Since  $\frac{p^k \cdot v}{(k-2)!(1-p)^{k-1}}$  and  $c$  are constant, and the decreasing rate of  $(1-p)^t$  is higher than the increasing rate of  $t$ , as  $t$  increases,  $t \cdot (1-p)^t \cdot \frac{p^k \cdot v}{(k-2)!(1-p)^{k-1}} - c$  will be less than or equal to 0 for a sufficiently large  $t$ . Thus, this proposition is proved. ■

**Proof of Theorem 2** We first prove that the extra cost expressed in Formula (10) is complete and non-redundant. Because we assume that node  $l$  ( $1 < l < m$ ) is the first failure node (receiving less than  $k$  packets), it enumerates all possible cases of failure. Also,

Formula (10) only enumerates the case for which the number of received packets  $t$  is less than  $k$ , since failure occurs only when node  $l$  receives less than  $k$  packets. Since Formula (10) recursively counts all the extra costs introduced at downstream nodes of node  $l$ , if node  $l$  fails, it exhaustively includes all possible extra costs in the event of the failure at node  $l$ . Hence the counted extra cost is complete and non-redundant. The optimality is thereby straightforward by following the proof of Dijkstra's shortest path algorithm since our model also satisfies the sub-path optimality, i.e., the sub-path of an optimal path is also optimal. It can be proved through simple contradiction. We omit it due to space limitation.

**Proof of Theorem 3** Similarly, for sub-path from  $j$  to  $d$  of path  $R$ , we define  $\Delta U_{jd}(t_j) = U_{jd}(t_j + 1) - U_{jd}(t_j)$ , which can be expressed as:

$$\Delta U_{jd}(t_j) = \sum_{t=k-1}^{t_j} \binom{t_j}{t} p^{t+1} (1-p)^{t_j-t} \Delta U_{j+1,d}(t) - c, \quad (16)$$

where  $p$  and  $c$  are the abbreviation of  $p_{j,j+1}$  and  $c_{j,j+1}$ , respectively.

*Lemma 2:* Formula (16) is correct.

*Proof:* In this proof, we use  $t'_j$  to denote  $t_j + 1$  to save space. According to Formula (13), we have

$$U_{jd}(t'_j) = \sum_{t=k}^{t'_j} \binom{t'_j}{t} p^t (1-p)^{t'_j-t} U_{j+1,d}(t) - t'_j \cdot c \quad (17)$$

Since  $\binom{t'_j}{t} = \binom{t'_j}{t-1} + \binom{t'_j}{t}$  (Pascal's rule), the right side of Formula (17) is

$$\sum_{t=k}^{t'_j} \left( \binom{t'_j}{t-1} + \binom{t'_j}{t} \right) p^t (1-p)^{t'_j-t} U_{j+1,d}(t) + p^{t'_j} U_{j+1,d}(t'_j) \quad (18)$$

minus  $t'_j \cdot c$ . Formula (18) can be rewritten as the sum of two formulas:  $\sum_{t=k}^{t'_j+1} \binom{t'_j}{t-1} p^t (1-p)^{t'_j-t+1} U_{j+1,d}(t)$  and  $\sum_{t=k}^{t'_j} \binom{t'_j}{t} p^t (1-p)^{t'_j-t} U_{j+1,d}(t)$ . The former can be rewritten as  $\sum_{t=k-1}^{t'_j} \binom{t'_j}{t} p^{t+1} \cdot (1-p)^{t'_j-t} U_{j+1,d}(t+1)$ , which is equal to  $\sum_{t=k-1}^{t'_j} \binom{t'_j}{t} p^{t+1} (1-p)^{t'_j-t} (U_{j+1,d}(t) + \Delta U_{j+1,d}(t))$  by replacing  $U_{j+1,d}(t+1)$  with  $U_{j+1,d}(t) + \Delta U_{j+1,d}(t)$ . Hence, we obtain  $\sum_{t=k-1}^{t'_j} \binom{t'_j}{t} p^{t+1} (1-p)^{t'_j-t} U_{j+1,d}(t) + \sum_{t=k-1}^{t'_j} \binom{t'_j}{t} p^{t+1} (1-p)^{t'_j-t} \Delta U_{j+1,d}(t)$ , which is equal to  $U_{jd}(t'_j) + \sum_{t=k-1}^{t'_j} \binom{t'_j}{t} p^{t+1} (1-p)^{t'_j-t} \Delta U_{j+1,d}(t)$  according to Formula (13) and the fact that  $U_{j+1,d}(k-1) = 0$ . Based on the above discussion, we have  $U_{jd}(t'_j) = U_{jd}(t_j) + \sum_{t=k-1}^{t'_j} \binom{t'_j}{t} p^{t+1} (1-p)^{t'_j-t} \Delta U_{j+1,d}(t) - c$ . Thus, this lemma is proved according to the definition of  $\Delta U_{jd}(t_j)$ . ■

*Proof:* We prove this by induction over the number of hops to the destination. For the inductive basis, a  $t_{m-1}^*$  exists such that  $\Delta U_{m-1,d}(t_{m-1}^* - 1) > 0$  and  $\Delta U_{m-1,d}(t_{m-1}^*) \leq 0$ , which can be directly derived from Theorem 1. By the inductive hypothesis, a  $t_{j-1}^*$  exists such that  $\Delta U_{j-1,d}(t_{j-1}^* - 1) > 0$  and  $\Delta U_{j-1,d}(t_{j-1}^*) \leq 0$ . Based on Lemma 2, for a sufficiently large  $t_j > t_{j-1}^*$ ,  $\Delta U_{jd}(t_j) < 0$ . Therefore, an optimal  $t_j^*$  exists. This theorem is then proved. ■