

On Optimal Diversity in Network-Coding-Based Routing in Wireless Networks

Qiao Xiang¹, Hongwei Zhang², Jianping Wang³, Guoliang Xing⁴, Shan Lin⁵, Xue Liu¹

¹School of Computer Science, McGill University, Canada

²Department of Computer Science, Wayne State University, United States

³Department of Computer Science, City University of Hong Kong, China

⁴Department of Computer Science and Engineering, Michigan State University, United States

⁵Department of Electrical and Computer Engineering, Stony Brook University, United States

qiao.xiang@mcgill.ca, hongwei@wayne.edu, jianwang@cityu.edu.hk,

glxing@cse.msu.edu, shan.x.lin@stonybrook.edu, xueliu@cs.mcgill.ca

Abstract—Network coding (NC) based opportunistic routing has been well studied, but the impact of routing diversity on the performance of NC-based routing remains largely unexplored. Towards understanding the importance of routing diversity in NC-based routing, we study the problems of estimating and minimizing the data delivery cost in NC-based routing. In particular, we propose an analytical framework for estimating the total number of packet transmissions for NC-based routing in arbitrary topologies. We design a greedy algorithm that minimizes the total transmission cost of NC-based routing and determines the corresponding forwarder set for each node. We prove the optimality of this algorithm and show that 1) nodes on the shortest path may not always be favored when selecting forwarders for NC-based routing and 2) the minimal cost of NC-based routing is upper-bounded by the cost of shortest path routing. Based on the greedy, optimal algorithm, we design and implement ONCR, a distributed minimal cost NC-based routing protocol. Using the NetEye sensor testbed, we comparatively study the performance of ONCR and existing approaches such as the single path routing protocol CTP and the NC-based opportunistic routing protocols MORE and CodeOR. Results show that ONCR achieves close to 100% delivery reliability while having the lowest delivery cost among all the protocols and 25-28% less than the second best protocol CTP. This low delivery cost also enables ONCR to achieve the highest network goodput, i.e., about two-fold improvement over MORE and CodeOR. Our findings demonstrate the significance of optimizing data forwarding diversity in NC-based routing for data delivery reliability, efficiency, and goodput.

I. INTRODUCTION

The past decade witnesses the fast advance of resource-constrained wireless communication systems, e.g., sensor networks. For resource-constrained wireless networks, many in-network processing (INP) methods have been proposed to improve energy efficiency and data delivery performance by reducing network traffic load and thus channel contention, and one such INP method is network coding. First proposed for wired networks [2], network coding (NC) improves network throughput by mixing packets at intermediate nodes during transmissions. As a special INP, it has drawn great interest from the wireless research community, and one important application scenario is network-coding-based routing. In particular, several NC-based opportunistic routing (OR) protocols [4][15][13] have been proposed. They improve overall throughput of wireless mesh networks through the integration of random network coding and opportunistic routing and corresponding rate control schemes.

When designing an NC-based routing protocols, there are usually three key challenges: 1) *how should a node select the forward set?* 2) *when should a node stop broadcasting re-encoded packets?* and 3) *when should a node start broadcasting re-encoded packets?* Various solutions to challenge 2 and 3 have been proposed in existing NC-based opportunistic routing protocols, e.g., [4][15][13]. Challenge 1, however, has been overlooked in existing protocols because they try to make full use of all routing diversity by allowing every node to take part in data transmissions. Unfortunately, as we will show later in this paper, this opportunistic approach may well lead to high data delivery cost (e.g., in terms of the number of packet transmissions taken) and thus are not suitable for resource-constrained wireless networks. Therefore, how to control routing diversity for minimum-cost NC-based routing remains largely unexplored so far. For instance, nodes on the shortest path routing are always selected into forwarder set in existing NC-based opportunistic routing protocols. However, as we will prove in Section V, this intuitive choice sometimes would increase the data delivery cost of NC-based routing.

In this paper, we investigate the impact of routing diversity on the cost of NC-based routing by exploring solutions to the cost estimation and cost minimization problems in NC-based routing. Different from existing NC-based OR protocols, which allow every node to forward coded data, our findings show that it is important to optimize routing diversity in NC-based routing. The **main contribution** of this paper is:

- We propose the first analytical framework for estimating the total number of packet transmissions of NC-based routing in arbitrary topologies;
- We propose a greedy algorithm that minimizes the total transmission cost of NC-based routing for arbitrary topologies and determines the corresponding forwarder set for each node. We prove its optimality and find an upper-bound for minimal cost NC-based routing, which equals to the cost of shortest single path routing. We also show that selecting nodes on the shortest path into forwarder set of NC-based routing may lead to worse delivery efficiency;
- We design and implement ONCR, a fully distributed minimal cost NC-based routing protocol, for resource-constrained sensor platforms. ONCR consists of a routing engine, a M-NSB coded feedback scheme, and a rate control module. The M-NSB scheme provides a precise coded feedback for NC-based forwarding, and the rate control module reduces the probability of linear dependency between coded packets due to link correlation;

- We evaluate the performance of ONCR on the NetEye sensor network testbed by comparing with CTP[6], MORE[4], and CodeOR[15]. Results show that ONCR achieves significantly better performance than all other protocols in terms of delivery reliability, efficiency, and goodput.

The remaining of this paper is organized as follows. We present the system model and problem definition in Section II. We propose an analytical framework for estimating the cost of NC-based routing in Section III. We design a greedy algorithm for minimizing the transmission cost in NC-based routing and prove its optimality in Section IV. We analyze the properties of minimal cost NC-based routing in Section V. We present the design of ONCR protocol in Section VI and evaluate its performance in Section VII. We discuss related work in Section VIII and make concluding remarks in Section IX.

II. SYSTEM MODEL AND PROBLEM DEFINITION

A. System model

We model a wireless network as a graph $G = (V, E)$ with a source S and a destination T . If S has K original information elements to transmit, the destination T will need to receive K linearly independent packets to decode. And we call this K the batch size. For each node $i \in G$, we use U_i and D_i to denote the set of 1-hop senders and receivers of i , respectively. And we denote the forwarder set of i as $FS_i \subset D_i$. For each link $(i, j) \in E$, we denote its expected number of transmissions to deliver a packet with payload length of l as ETX_{ij}^l and define $P_{ij}^l = \frac{1}{ETX_{ij}^l}$ as the corresponding link reliability. Because NC-based routing does not change the packet payload length, we use ETX_{ij} and P_{ij} in the remaining of this paper for simplicity. Then we define $C_{ij}(x)$ as the transmission cost of delivering x linearly independent packets from i to j . When $x = 1$, we use C_{ij} for simplicity. And it is straightforward to see $C_{ij}(x) = x \cdot C_{ij}$. We also define $C_{i \rightarrow FS_i}(x)$ as the expected number of broadcasts of node i when nodes in FS_i collectively receive x linearly independent coded packets from i . We use a large finite field size in generating network coding coefficient, e.g., 2^8 , 2^{16} or 2^{32} , so that the probability of two randomly coded packets generated by the same node being linearly dependent approaches zero [9]. In order for the problem to be tractable, we assume different links that share the same sender are independent with each other [17]. In Section VI, we design a rate control module for our ONCR protocol to relax this assumption. And we show in Section VII that the algorithm developed based on this assumption also leads to improved data forwarding performance when links may be correlated, i.e., in the NetEye testbed.

B. Problem definition

Given the system model above, we first define the cost estimation problem of NC-based routing as follows:

EST-NC Problem: given a network G and K information elements originated at source S , estimate the total number of packet transmissions of NC-based routing to deliver K linearly independent packets to T where $FS_i, \forall i$ is known *a priori*.

The solution to **EST-NC** problem will provide an analytical framework to compute the expected transmission cost of different NC-based routing protocols. Based on this framework, we define the cost minimization problem of NC-based routing:

MIN-NC Problem: given a network G and K information elements originated at source S , find the minimal total number of packet transmissions of NC-based routing and the corresponding FS_i for each node $i \in G$ to deliver K linearly independent packets to T .

III. AN ANALYTICAL FRAMEWORK FOR NC-BASED ROUTING

Existing NC-based routing protocols, such as MORE [4] and CodeOR [15], can significantly improve network throughput. However, the transmission cost of these protocols, i.e., total number of packet transmissions, can be much higher than shortest single path routing because they utilize all routing diversity in protocol design, where each intermediate node takes part in the forwarding process by broadcasting re-encoded packets to its own forwarder candidate set. These protocols [4][15][13] adopt different credit computation algorithms to control redundant transmissions of each node. However, the total number of packet transmissions in NC-based routing protocols has not been studied. In this section, we propose a load assignment algorithm to solve the **EST-NC** problem, which provides the first analytical framework to estimate the total number of packet transmissions of NC-based routing protocols. As the first step of solving **EST-NC** problem, we now discuss the solution of the following simpler problem.

D-EST-NC problem: the same as **EST-NC** problem except that there are only two types of links in G : 1)links from S to every intermediate node A_i and 2)links from every intermediate node A_i to T .

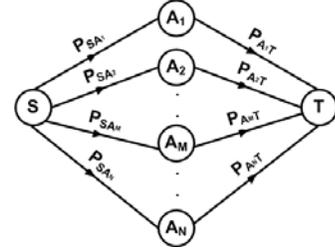


Fig. 1. **D-EST-NC** problem: diamond topology

The diamond topology in **D-EST-NC** problem is shown in Figure 1. We assume $|FS_S| = M, |D_S| = N$ and $M \leq N$. And it is easy to see that for each link $A_i \rightarrow T$, $C_{A_i T} = ETX_{A_i T} = \frac{1}{P_{A_i T}}$. In this problem, the cost to deliver K linearly independent packets from S to T can be decomposed into two parts. The first part is the cost of one-hop broadcast from S to FS_S and the second part is the forwarding cost from FS_S to T . The one-hop broadcast cost from S to FS_S is the expected number of broadcast for nodes in FS_S to collectively receive K linearly independent packets. Estimating this cost requires us treating the forwarder set FS_S as one virtual node V_s . The reliability of the corresponding virtual link $S \rightarrow V_s$ is then expressed as $P_{SV_s} = 1 - \prod_{i=1}^M (1 - P_{SA_i})$ under the link independence assumption. Therefore, the transmission cost for one-hop broadcast can be expressed as:

$$C_{S \rightarrow FS_S}(K) = \frac{K}{1 - \prod_{i=1}^M (1 - P_{SA_i})} \quad (1)$$

To estimate the forwarding cost from FS_S to T , we need to determine the number of encoded packets each intermediate node should forward towards T . To tackle this challenge, we propose a concept called *effective load*.

Definition 1: Given a batch n , for a node j in the forwarder candidate set FS_i , the **effective load** $L_j^i(n)$ is defined as the number of linearly independent packets of batch n sent by i that are received by j but not by any of the other nodes in FS_i that has lower transmission cost to the destination.

Effective load represents the entropy j can contribute in decoding batch n , i.e., the number of coded packets j can

deliver to T which are linearly independent to packets delivered by other nodes in FS_i . Therefore, node j only needs to forward $L_j^i(n)$ coded packets to T . This concept enables us to precisely estimate the forwarding cost of different nodes within the same forwarder set. Because the **D-EST-NC** problem only has one batch of K information elements, we use L_j^i for simplicity. Without loss of generality, we assume that $C_{A_1T} \leq C_{A_2T}, \dots, \leq C_{A_MT}$ in **D-EST-NC** problem. Based on the definition, the effective load $L_{A_i}^S$ can be computed as

$$L_{A_i}^S = C_{S \rightarrow FS_S}(K) \times P_{SA_i} \times \prod_{j=1}^{i-1} (1 - P_{SA_j}) \quad (2)$$

And we propose the following lemma:

Lemma 1: For any instance of **D-EST-NC** problem, we have $\sum_{i=1}^M L_{A_i}^S = K$.

The deduction of Equation 2 and proof of Lemma 1 are to solve some basic probability theory problems and are omitted due to the constraint of space. Readers may refer to technical report [20] for more details. Lemma 1 indicates that in order for node T to receive K linearly independent packets, each intermediate node A_i only needs to forward $L_{A_i}^S$ re-encoded packets to T . Any additional coded packets forwarded to T will be redundant because they are linearly dependent to these K packets. These redundant packet forwarding therefore should not be included when estimating the transmission cost of NC-based routing. With this result, we can estimate the forwarding cost from FS_S to T . By summing up the one-hop broadcast cost in Equation 1 and the forwarding cost of FS_S , we express the total number of transmissions to deliver K linearly independent packets from S to T as:

$$\begin{aligned} C_{ST}(K) &= C_{S \rightarrow FS_S}(K) + \sum_{i=1}^M L_{A_i}^S C_{A_iT} \\ &= \frac{K}{1 - \prod_{i=1}^M (1 - P_{SA_i})} \cdot \left\{ 1 + \sum_{i=1}^M [C_{A_iT} \cdot P_{SA_i} \cdot \prod_{j=1}^{i-1} (1 - P_{SA_j})] \right\} \end{aligned} \quad (3)$$

Algorithm 1 Two-step algorithm for **D-EST-NC** problem

- 1: Input: a diamond network G with source S , $FS_S = \{A_1, A_2, \dots, A_M\}$
 - 2: Output: $C_{ST}(K)$: the expected number of transmissions to deliver K linearly independent packets from S to T
 - 3: Sort nodes in FS_S by a non-descending order of C_{A_iT} , where $i = 1, 2, \dots, M$.
 - 4: Sorted nodes are labeled as $\{A'_1, A'_2, \dots, A'_M\}$
 - 5: $C_{S \rightarrow FS_S}(K) = \frac{K}{1 - \prod_{i=1}^M (1 - P_{SA'_i})}$
 - 6: $L_{A'_1}^S = C_{S \rightarrow FS_S}(K) P_{SA'_1}$
 - 7: $F = 1 - P_{SA'_1}$
 - 8: **for** $i \rightarrow 2, 3, \dots, M$ **do**
 - 9: $L_{A'_i}^S = C_{S \rightarrow FS_S}(K) P_{SA'_i} F$
 - 10: $C_{A'_i}(L_{A'_i}^S) = L_{A'_i}^S C_{A'_iT}$
 - 11: $F = F(1 - P_{SA'_i})$
 - 12: **end for**
 - 13: $C_{ST}(K) = C_{S \rightarrow FS_S}(K) + \sum_{i=1}^M C_{A'_i}(L_{A'_i}^S)$
-

Equation 3 gives a close-form solution to the **D-EST-NC** problem. We formally present this computing process in Algorithm 1. This algorithm estimates the cost of broadcast by treating the forwarder set as a virtual node, estimates the cost of forwarding by computing the effective load of each forwarder, and adds two parts together to get the final result. Note that in Algorithm 1, we sort nodes of FS_S by a non-descending order of their transmission cost to T before computing the total transmission cost. This step relaxes the assumption of ordered forwarder set in our deduction above. The time complexity of Algorithm 1 is $O(|V| \lg |V|)$.

After solving **D-EST-NC** problem, we generalize this effective load based approach to recursively estimate the total number of packet transmissions of NC-based routing in arbitrary topologies. Given an instance of **EST-NC** problem, we execute Algorithm 1 for each non-root i to compute $C_{iT}(K)$ if every node j in FS_i has computed and updated $C_{jT}(K)$. We then update $C_{iT}(K)$ to notify U_i , the set of i 's one-hop senders. We execute the same procedure for nodes in U_i to update their transmission cost to the destination. Because for each upstream sender $i \in U_j$, node j only forwards the corresponding effective load L_j^i to FS_j , the probability of two encoded packets sent by different forwarders but received by the same next-hop forwarder being linearly dependent approaches 0 with a large finite field size. This property ensures that this backwards recursive solution process is applicable to arbitrary topologies. Upon the convergence of this backwards recursive process, we will be able to estimate the total number of packet transmissions to deliver K linearly independent packets from S to T . Therefore, we have a solution to **EST-NC** problem.

Discussion. This analytical framework can be applied to estimate the transmission cost for different combinations of forwarder sets, including NC-based opportunistic routing protocols in which $FS_i = D_i$ for every non-root node i . It can be run at each node in the network as a distance-vector algorithm. We discuss loop avoidance mechanism for this framework in Section VI. A credit assignment algorithm with similar idea of Algorithm 1 was also proposed in [11] for physical-layer coding based opportunistic routing but its objective is to reduce the queuing delay. In the next few sections, we will show that selecting the optimal forwarder set to minimize the total number of packet transmissions of NC-based routing is a crucial yet non-trivial problem for resource-constrained wireless networks.

IV. OPTIMIZING DIVERSITY OF NC-BASED ROUTING

Last section we propose an analytical framework to estimate the transmission cost of NC-based routing in arbitrary topologies with arbitrary forwarder sets. However, without any control on forwarder selection, the total number of packet transmissions of NC-based routing could be extremely high, reducing the delivery efficiency and impairing network performance of wireless networks. This phenomenon is especially severe in resource-constrained mission-critical sensor networks. In this section, we design a greedy algorithm that determines the forwarder set of each non-root node in wireless networks to get an optimal solution to the **MIN-NC** problem.

In Section III, we show that by running Algorithm 1 for each node in a backwards recursive way, we can estimate the total number of packet transmissions of NC-based routing in arbitrary topologies. Similarly, we can solve the **MIN-NC** problem by solving a simple version first and then use the same backwards recursive strategy to get the final solution. The simple version of **MIN-NC** problem we start with is

D-MIN-NC problem The same as the **MIN-NC** problem except that there are only two types of links in G : 1) links from S to every intermediate node A_i and 2) links from every intermediate node A_i to T .

It can be seen that the **D-MIN-NC** problem has the same diamond topology as **D-EST-NC** problem. To solve this problem, we design a greedy algorithm which is presented as Algorithm 2. The basic idea of Algorithm 2 is to first sort all nodes in D_S in a non-descending order of their transmission cost to the destination. We then remove the node A'_i , which has the lowest transmission cost to T , from the sorted D_S , add it

to the forwarder set FS_S and compute the total transmission cost using Algorithm 1. If the total transmission cost of S can be reduced by adding A'_i to FS_S , we keep it in FS_S and add another node with the lowest cost from the remaining sorted D_S . We continue this selection process until either of the following two conditions is satisfied:

- the sorted D_S is empty, i.e., all one-hop receivers of S have been selected into FS_S ;
- moving another node from the sorted D_S to FS_S would increase the total transmission cost of S .

Algorithm 2 has a time complexity of $O(|V|^2 \lg |V|)$. Not only will it provide the minimal total transmission cost and forwarder set, it also determine the effective load that should be assigned to each forwarder. When this algorithm terminates, it is possible that $|FS_S| = 1$. In this case the optimal routing structure of NC-based routing is the shortest single path.

To show this greedy algorithm provides the optimal solution to the **D-MIN-NC** problem, we propose and prove the following theorem:

Theorem 1: Given an instance of **D-MIN-NC** problem with source S and $D_S = \{A_1, A_2, \dots, A_M\}$, Algorithm 2 yields the minimal total transmission cost $C_{ST}^*(K)$ and the corresponding forwarder set FS_S for NC-based routing.

Proof: We prove this theorem by contradiction. Denote the minimal transmission cost to deliver K linearly independent coded packets from S to T as $C_{ST}^*(K)$ and the corresponding forwarder set as FS_S^* with $|FS_S^*| = k$. We sort nodes in FS_S^* in non-descending order of their transmission cost to the destination T and denote them as $FS_S^* = \{A_1^*, A_2^*, \dots, A_k^*\}$ with $C_{A_1^*T} \leq C_{A_2^*T} \leq \dots \leq C_{A_k^*T}$.

If this theorem does not hold, there exists at least one node A_x that is not selected into FS_S^* and has $C_{A_xT} < C_{A_i^*T}$ for some integer $i \in [1, k]$. Without loss of generality, we assume that $C_{A_{k-1}^*T} < C_{A_xT} < C_{A_k^*T}$. We will have a contradiction when we can find a forwarder set FS_S^{**} that has a lower transmission cost C_{ST}^{**} than C_{ST}^* . To find this contradiction, we study the three forwarder sets, i.e., $FS_S^* = \{A_1^*, A_2^*, \dots, A_k^*\}$, $FS_S^1 = FS_S^* - \{A_k^*\}$, and $FS_S^2 = FS_S^* \cup \{A_x\}$.

Algorithm 2 Greedy algorithm for **D-MIN-NC** problem

- 1: Input: node S , $D_S = \{A_1, A_2, \dots, A_M\}$, $FS_S = \emptyset$
 - 2: Output: $C_{ST}^*(K)$: the minimal transmission cost to deliver K linearly independent packets from S to T and the corresponding forwarder set FS_S
 - 3: Sort nodes in D_S by a non-descending order of C_{A_iT} , where $i = 1, 2, \dots, M$.
 - 4: Sorted nodes are labeled as $\{A'_1, A'_2, \dots, A'_M\}$
 - 5: $FS_S = \{A'_1\}$
 - 6: $C_{ST}^*(K) = \frac{1}{P_{SA'_1}} + K \cdot C_{A'_1T}$
 - 7: **for** $i \rightarrow 2, 3, \dots, M$ **do**
 - 8: Run Algorithm 1 with input S and $D_S = \{A'_1 \dots A'_i\}$
 - 9: Get the result as $C_{ST}^{new}(K)$
 - 10: **if** $C_{ST}^{new}(K) > C_{ST}^*(K)$ **then**
 - 11: break
 - 12: **else**
 - 13: $FS_S = FS_S \cup A'_i$
 - 14: $C_{ST}^*(K) = C_{ST}^{new}(K)$
 - 15: **end if**
 - 16: **end for**
-

For each forwarder set, we compute the corresponding total transmission cost using Algorithm 1. The results are shown as

$$C_{ST}^*(K) = K \cdot \frac{1 + \sum_{i=1}^k [C_{A_i^*T} P_{SA_i^*} \prod_{j=1}^{i-1} (1 - P_{SA_j^*})]}{1 - \prod_{i=1}^k (1 - P_{SA_i^*})} \quad (4)$$

$$C_{ST}^1(K) = K \cdot \frac{1 + \sum_{i=1}^{k-1} [C_{A_i^*T} P_{SA_i^*} \prod_{j=1}^{i-1} (1 - P_{SA_j^*})]}{1 - \prod_{i=1}^{k-1} (1 - P_{SA_i^*})} \quad (5)$$

$$C_{ST}^2(K) = \frac{1 - (1 - P_{SA_x}) \prod_{i=1}^k (1 - P_{SA_i^*})}{1 - (1 - P_{SA_x}) \prod_{i=1}^{k-1} (1 - P_{SA_i^*})} \cdot \{1 + \sum_{i=1}^{k-1} [C_{A_i^*T} P_{SA_i^*} \prod_{j=1}^{i-1} (1 - P_{SA_j^*})] + C_{A_xT} P_{SA_x} \prod_{i=1}^{k-1} (1 - P_{SA_i^*}) + C_{A_k^*T} P_{SA_k^*} (1 - P_{SA_x}) \prod_{i=1}^{k-1} (1 - P_{SA_i^*})\} \quad (6)$$

Since $C_{ST}^*(K)$ is the minimal total transmission cost, we have $C_{ST}^*(K) \leq C_{ST}^1(K)$. Through some mathematical transformation on this inequity, we get a useful result in Inequity 7:

$$1 + \sum_{i=1}^{k-1} [C_{A_i^*T} P_{SA_i^*} \prod_{j=1}^{i-1} (1 - P_{SA_j^*})] \geq [1 - \prod_{i=1}^{k-1} (1 - P_{SA_i^*})] C_{A_k^*T} \quad (7)$$

Next, we compute $C_{ST}^*(K) - C_{ST}^2(K)$. After some mathematical transformation, we can get the results in Inequity 8

$$C_{ST}^*(K) - C_{ST}^2(K) = \frac{P_{SA_x} \prod_{i=1}^{k-1} (1 - P_{SA_i^*})}{[1 - \prod_{i=1}^k (1 - P_{SA_i^*})]} \cdot \frac{[1 - (1 - P_{SA_x}) \prod_{i=1}^{k-1} (1 - P_{SA_i^*})]}{K} \cdot \{ (1 - P_{SA_k^*}) \{ 1 + \sum_{i=1}^{k-1} [C_{A_i^*T} P_{SA_i^*} \prod_{j=1}^{i-1} (1 - P_{SA_j^*})] \} - [1 - \prod_{i=1}^k (1 - P_{SA_i^*})] C_{A_xT} \} + P_{SA_k^*} C_{A_k^*T} - P_{SA_k^*} C_{A_xT} \} \quad (8)$$

Leveraging the result of Inequity 7 and the fact that $C_{A_k^*T} > C_{A_xT}$, we can find that the right hand side of Inequity 8 is greater than 0, i.e., $C_{ST}^*(K) > C_{ST}^2(K)$. This result contradicts with the assumption that $C_{ST}^*(K)$ is the minimal total transmission cost. It is also straightforward that using the above mathematical deduction framework, a contradiction can be found for any choice of i and x where $i \in [1, k]$ and $C_{A_i^*} > C_{A_x}$. Therefore, we prove that we can always find the minimal total transmission cost from S to T by selecting A_i with lowest C_{A_iT} from D_S into FS_S and stopping when adding more candidates would increase $C_{ST}(K)$. In the end, we complete our proof on the optimality of Algorithm 2 in solving **D-MIN-NC** problem. ■

Having solved the **D-MIN-NC** problem, we can use the same backwards recursive approach in Section III to solve **MIN-NC** in arbitrary topologies. Given an instance of **MIN-NC** problem, we execute Algorithm 2 for each non-root i to compute $C_{iT}^*(K)$ if every node j in FS_i has computed and updated $C_{jT}^*(K)$. We then update $C_{iT}^*(K)$ to notify U_i , the set of i 's one-hop senders. We then run Algorithm 2 for nodes in U_i to update their transmission cost to the destination. Upon the convergence of this backwards recursive process, we will be able to determine the minimal total number of packet transmissions to deliver K linearly independent packets from S to T and the corresponding forwarder set for each non-root node. To this end, we have a solution to **MIN-NC** problem.

Discussion. Similar as **EST-NC** problem, our solution to **MIN-NC** problem is applicable to arbitrary topologies and its optimality is not affected by the overlapping of different forwarder sets due to the concept of effective load and the large finite field size used for random network coding, e.g., 2^8 , 2^{16} and 2^{32} . The backwards recursive approach for **MIN-NC** can also be implemented as a distance-vector protocol. We discuss implementation details including loop-avoidance in Section VI. Detailed mathematical transformation in proving Theorem 1 is omitted due to the constraint of space and can be found in the technical report [20].

V. NC-BASED ROUTING VS. SHORTEST PATH ROUTING: A THEORETICAL COMPARISON

The greedy algorithm we propose in Section IV yields the minimal total transmission cost of NC-based routing. When implementing a routing protocol for wireless networks, nonetheless, we still need to face the choice between NC-based routing and shortest single path routing. The potential benefits of utilizing NC-based routing in wireless networks, especially on resource-constrained sensor platforms, are of great importance in choosing routing strategies. In this section, we conduct a theoretical comparison on the total transmission cost between NC-based routing and shortest single path routing by exploring properties of Algorithm 2. Our findings demonstrate the advantage of minimal cost NC-based routing over shortest path routing.

The first property we find from Algorithm 2 is that the non-terminal nodes on the shortest (i.e., lowest cost) single path is not necessarily chosen in the optimal forwarder set. This property is formally presented in the following theorem:

Theorem 2: Given a graph G with source node S and its one-hop candidate set D_S of M forwarders, the optimal forwarder set FS_S computed in Algorithm 2 does not always contain node A^* where $A^* \in D_S$ and A^* is on the shortest single path from S to T .

Proof: We only need an instance of **MIN-NC** problem whose optimal NC-based routing topology does not include the shortest single path to prove this theorem. Thus we build an instance in Figure 2. Numbers in this figure represents reliability of different links.

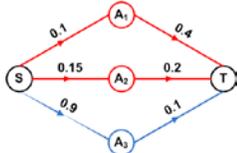


Fig. 2. A NC-based route without shortest single path

In this instance, the shortest single path is $S \rightarrow A_3 \rightarrow T$ which is marked blue in Figure 2 and its total transmission cost to delivery one packet is $\frac{1}{0.9} + \frac{1}{0.1} = 11.11$. After we execute Algorithm 2, however, the optimal forwarder set we have is $FS_S = \{A_1, A_2\}$, which is marked red in the figure, because $C_{S \rightarrow \{A_1, A_2\} \rightarrow T}(1) = 8.1915$ while $C_{S \rightarrow \{A_1, A_2, A_3\} \rightarrow T}(1) = 9.5398$. With this instance, we complete our proof. ■

In this example, we can see that the minimal transmission cost of NC-based routing is lower than that of shortest single path. This observation further raises the question: will the minimal cost of NC-based routing always be lower than that of shortest single path? To answer this question, we propose the second property of Algorithm 2 in the following theorem:

Theorem 3: Given a graph G with source node S and its one-hop candidate set D_S of M forwarders, the optimal transmission cost $C_{ST}^*(K)$ computed by Algorithm 2 is upper bounded by $C_{single}^*(K)$ where $C_{single}^*(K)$ is the cost of shortest single path from S to T .

Proof: It is shown in previous sections that the correctness of our solutions to **EST-NC** and **MIN-NC** problems will not be affected by the overlapping between forwarder sets of different nodes. Therefore we only need to prove the correctness of this theorem in diamond topology used in **D-EST-NC** and **D-MIN-NC** problems. It then can be naturally extended to arbitrary topologies. In the diamond topology, we have $C_{A_i T} = \frac{1}{P_{A_i T}}$ for each link $A_i \rightarrow T$. We assume the shortest single path is $S \rightarrow A^* \rightarrow T$ and we have

$C_{single}^*(K) = K(\frac{1}{P_{SA^*}} + C_{A^* T})$ Using the finding in Theorem 2, we prove this theorem in diamond topology under two different cases:

1) $A^* \notin FS_S$. When the forwarder A^* on the lowest cost single path is not selected into FS_S , based on the greedy principle adopted by Algorithm 2, we have $C_{A^* T} \geq C_{A_i T}$ for any $A_i \in FS_S$. Denoting $FS_S = \{A_1, A_2, \dots, A_k\}$, we make some transformation on $C_{ST}^*(K) - C_{ST}^{FS_S \cup \{A^*\}}(K)$ and we get:

$$\begin{aligned} & C_{ST}^*(K) - C_{ST}^{FS_S \cup \{A^*\}}(K) \\ &= \frac{K \cdot P_{SA^*} \prod_{i=1}^k (1 - P_{SA_i})}{1 - (1 - P_{SA^*}) \prod_{i=1}^k (1 - P_{SA_i})} \\ & \cdot \left\{ \frac{1 + \sum_{i=1}^k [C_{A_i T} P_{SA_i} \prod_{j=1}^{i-1} (1 - P_{SA_j})]}{1 - \prod_{i=1}^k (1 - P_{SA_i})} - C_{A^* T} \right\} < 0 \end{aligned} \quad (9)$$

From this inequality, we then have:

$$\begin{aligned} & K \cdot \left\{ \frac{1 + \sum_{i=1}^k [C_{A_i T} P_{SA_i} \prod_{j=1}^{i-1} (1 - P_{SA_j})]}{1 - \prod_{i=1}^k (1 - P_{SA_i})} - C_{A^* T} \right\} < 0 \\ & \Leftrightarrow C_{ST}^*(K) - C_{A^* T}(K) < 0 \\ & \Rightarrow C_{ST}^*(K) < K(\frac{1}{P_{SA^*}} + C_{A^* T}) \end{aligned} \quad (10)$$

Therefore, we proved this theorem holds when A^* is not selected into FS_S .

2) $A^* \in FS_S$. In this case, there are three scenarios:

- If $FS_S = \{A^*\}$, it is clear that $C_{ST}^*(K) = K(\frac{1}{P_{SA^*}} + C_{A^* T})$.
- If $FS_S \neq \{A^*\}$ and A^* is the first node selected into FS_S , $C_{ST}^*(K) \leq K(\frac{1}{P_{SA^*}} + C_{A^* T})$ is implied in the greedy principle of Algorithm 2.
- If $FS_S \neq \{A^*\}$ and A^* is not the first node selected into FS_S , it is straightforward that

$$\frac{K}{1 - (1 - P_{SA^*}) \prod_{i \in FS_S^{old}} (1 - P_{SA_i})} < \frac{K}{P_{SA^*}} \quad (11)$$

where FS_S^{old} is the forwarder set before adding A^* . And it is implied in Algorithm 2 that before adding A^* into FS_S , $C_{A^* T}$ is greater than or equal to the forwarding cost of FS_S^{old} . Summing up the one-hop broadcast cost and the forwarding cost, we still have $C_{ST}^*(K) < K(\frac{1}{P_{SA^*}} + C_{A^* T})$ under this scenario because after adding A^* into forwarder set, the greedy algorithm may add more candidates into the optimal FS_S .

Combining all three scenarios, we prove this theorem holds when A^* is selected into FS_S . Therefore this theorem is correct under diamond topology. Since the large finite field size used by NC-based routing ensures that the probability of two encoded packets sent by different forwarders but received by the same next-hop forwarder being linearly dependent is close to zero, this proof framework is also applicable to arbitrary topologies, which completes our proof. ■

Discussion. Through Theorem 2 and 3, we prove that *nodes on the shortest path route should not always be favored in forwarder selection for NC-based routing*, which is different from opportunistic routing. This crucial property of NC-based routing has been neglected in all existing NC-based routing protocols. Moreover, this property shows the importance of carefully selecting forwarder set for NC-based routing. We also demonstrate the potentials of NC-based routing in improving data delivery efficiency in terms of total number of transmissions compared to shortest single path routing. This makes NC-based routing a desirable candidate for wireless networks, especially in resource-constrained sensor platforms.

VI. ONCR: A MINIMAL COST NC-BASED ROUTING PROTOCOL

Based on Algorithm 2, we design and implement ONCR, a minimal cost NC-based routing protocol for resource-constrained sensor networks. ONCR is a fully distributed routing protocol that runs on every node in the network. As discussed in Section I, there are usually three key challenges in designing an NC-based routing protocol. For a given batch,

1. How should a node select the forwarder set?
2. When should a node stop broadcasting re-encoded packets?
3. When should a node start broadcasting re-encoded packets?

In ONCR, we implement three components, i.e., routing engine, M-NSB ACK scheme and rate control module. Each of them addresses one challenge listed above.

A. Routing engine

As the core of ONCR, the routing engine is responsible in computing the optimal forwarder set for each node, which addresses the first challenge. In our design, we adopt the 4-bit link estimator of the collection tree protocol (CTP) [6] to provide single link reliability information. After getting link and route information from one-hop neighbors of current node, the routing engine performs the following tasks:

- Compute and update the minimal transmission cost of NC-based routing from current node to destination and the corresponding forwarder set;
- Broadcast the computed total transmission cost, the optimal forwarder set and the corresponding normalized effective load information to both one-hop receiver set and one-hop sender set of the current node;

The route and forwarder set computation process is performed by executing Algorithm 2. We forbid a node from being selected into forwarder set if its transmission cost to the destination is higher than the current node to avoid routing loops. This loop avoidance mechanism has been shown effective in wireless sensor networks [6]. It is easy to see that in a distributed environment, each node only needs to solve **D-MIN-NC** problem. And by broadcasting local transmission cost information to one-hop neighbors, a global optimal solution to the **MIN-NC** problem is achieved upon convergence. In this way, we optimally utilize wireless routing diversity to minimize the total number of packet transmissions of NC-based routing. Furthermore, since we compute a normalized effective load information in percentage, the routing engine is adaptive to arbitrary batch sizes.

B. Modified NSB coded feedback

In ONCR, the routing engine decides the forwarder set for the current node. However, in NC-based routing, each node also needs to know when it can stop broadcasting encoded packets of a batch to its forwarder set. The condition for a node i to stop broadcasting coded packets for a batch is that nodes in FS_i have collectively received $L_i(n)$ linearly independent packets, where $L_i(n)$ is the effective load of batch n assigned to i . This information is computed by routing engine in the one-hop sender of i and sent to i in encoded packets.

Besides $L_i(n)$, node i also needs feedback information, i.e., ACK, from nodes in FS_i . However, explicit ACK for broadcast is infeasible and an alternative approach is to use coded feedback. First proposed in [16], the null-space-based (NSB) coded feedback scheme is designed to enhance reliability of an NC-based multicast protocol for multimedia applications in mobile ad hoc networks. To apply coded

feedback into NC-based opportunistic routing, a Coded Cumulative ACK (CCACK) is proposed in [13]. CCACK designs a more complex ACK generating and testing scheme to solve the collective-space problem and false-positive problem when directly applying NSB in NC-based opportunistic routing. However, CCACK is originally designed to deploy in mesh networks, where nodes has stronger computation power and much larger storage space, making it infeasible to be transplanted to resource-constrained sensor platforms.

Motivated by NSB and CCACK, we propose M-NSB, a modified NSB ACK scheme for ONCR to address the second challenge. The false-positive problem does not exist in the design of ONCR, thus we only need to take care of the collective-space problem in our ACK design. NSB scheme is not designed to convey the collective space of all downstream nodes but only the space relationship between the individual node pairs. It generates coded ACK only based on coding vectors of received packet and this is why collective-space problem may arise. In our M-NSB ACK scheme, on the contrary, we generate coded ACK based on coding vectors of both received packets and forwarded packets. This design principle is similar as CCACK does and can solve the collective-space problem. However, we do not implement multiple checking matrices which are needed in CCACK to solve false-positive problem. In this way, M-NSB is able to provide a precise coded feedback with both low computation and storage overhead.

After an M-NSB ACK of a certain batch is generated at a receiving node j , it is broadcast to U_j , the set of one-hop sender of node j . We point out that M-NSB is also different from CCACK in that it does not take nodes overhearing from different upstream nodes into account. This is for the objective of precisely measuring and controlling the total transmission cost for the whole network. In ONCR, node j is assigned different effective load by different one-hop senders from U_j , and packets received by the same node but from different senders will be viewed as different batches, i.e., different traffic flows. By solving the collective-space problem for each sender separately, every coded packet can be effectively used for the decoding at the destination and there will be no redundant packet forwarding. To this end, M-NSB addresses the second design challenge for NC-based routing protocols.

C. Rate control

In ONCR, the routing engine component provides the effective load information, and the M-NSB component provides the receipt status of coded packets in the forwarder set. However, Algorithm 2 is designed based on the assumption of independence between broadcast links. This assumption may not always hold in reality and may cause high linearly dependence between received coded packets, which could impair the network performance. To fill this gap between theoretical analysis and real-world implementation, we design a simple rate-control component in ONCR to help each node decide when to start broadcasting re-encoded packets.

For a certain batch n broadcast from i to FS_i , our rate control component at nodes j where $j \in FS_i$ decide when j should start broadcasting re-encoded packets of batch n based on two parameters: the expected number of packets of j will receive from i , denoted by $E_j^i(n)$ and the effective load $L_j^i(n)$. Both parameters are computed by i and broadcast to FS_i . A batch n is ready for broadcasting at non-root node j only when the following condition is satisfied:

- Node j receives more than $E_j^i(n) - L_j^i(n)$ linearly independent packets from node i

We use this condition to relax the assumption of link independence in theoretical analysis. It can also prevent premature forwarding, i.e., forwarders with higher transmission cost to destination broadcasting encoded packets that are possibly linearly dependent with packets broadcast by forwarders with lower transmission cost. Meanwhile, this condition reduces the contention in the network by having nodes broadcast at different time. Based on this condition, when j has the highest priority in FS_i in terms of transmission cost to the destination, we have $E_j^i(n) = L_j^i(n)$ and batch n is ready for broadcasting at j as soon as the first packet is broadcast from i to j . And for every node, one ready-to-forward-batch is chosen for broadcast in a round-robin fashion every time it has a transmit opportunity. A re-encoded packet is generated by selecting non-zero elements in $GF(2^8)$ as re-encoding coefficients. Therefore, our rate-control model solves the third challenge for NC-based routing protocol.

VII. PERFORMANCE EVALUATION

To characterize the feasibility and effectiveness of NC-based routing in reducing total transmission cost in resource-constrained wireless networks, we implement ONCR on TelosB sensor platforms and experimentally evaluate its performance. We first present the experimentation methodology and then measurement results.

A. Methodology

Testbed. We use the *NetEye* sensor network testbed at Wayne State University [1]. 130 TelosB motes are deployed in *NetEye*, where every two closest neighboring motes are separated by 2 feet in an indoor environment. On each TelosB mote, a 3dB signal attenuator and a 2.45GHz monopole antenna are installed. In our measurement study, we set the radio transmission power to be -7dBm (i.e., power level 15 in TinyOS) such that a multihop network can be created. And we use the default MAC protocol provided in TinyOS 2.x.

Topology. During evaluation, we focus on data collection, i.e., convergecast traffic, since it is the most popular traffic scenario in wireless sensor networks. Out of the 130 motes in *NetEye*, we randomly select 40 motes (with each mote being selected with equal probability) to form a random network for our experimentation. Among these 40 motes, one node is selected as the data sink.

Protocol studied. To understand the importance of forwarder set selection for NC-based routing and its impact in improving the total transmission cost of wireless sensing and control, we comparatively study the following protocols:

- *ONCR*: the minimal cost NC-based routing protocol proposed in Section VI, which selects the optimal forwarder set for each node to minimize the total transmission cost;
- *CTP*: a state-of-the-art collection tree protocol designed for data collection in sensor networks [6];
- *MORE*: the first NC-based opportunistic routing protocol in wireless mesh networks, which utilizes full routing diversity in network [4];
- *CodeOR*: another representative NC-based opportunistic routing protocol in wireless mesh networks, which adds hop-by-hop ACK to the prototype of *MORE* and thus increases the concurrency of data flow [15].

Among existing NC-based OR protocols, *MORE* and *CodeOR* represent the major mechanisms that have been used to utilize the routing diversity in wireless networks. Other

protocols, e.g., *CCACK*[13] and *MIXIT*[11], are not expected to perform better than the ones we have considered in this study in terms of transmission cost. We implement all four protocols in TinyOS 2.x. For *ONCR*, *MORE* and *CodeOR*, we use finite field $GF(2^8)$ and batch size 8 for network coding operation because the length of information element in sensor network applications is usually short. We set the maximal size of forwarder set as 5 for all three NC-based routing protocols.

Performance metrics. For each protocol we study, we evaluate its behavior based on the following metrics:

- *Delivery reliability*: percentage of information elements correctly received by the sink;
- *Delivery cost*: number of transmissions required for delivering an information element from its source to the sink;
- *Goodput*: number of valid information elements received by the sink per second;
- *Routing diversity*: the size of forwarder set to transmit a batch.

When measuring goodput, we define an information element as **valid** if and only if it is linearly independent to all elements that are in the same batch and received by the sink. We do not study the routing diversity of *CTP* because it is a single path routing protocol and the routing diversity is always one.

Traffic pattern. In our experiment, we use two periodic data collection traffic patterns to represent light and heavy traffic scenarios, respectively.

- *S10*: 10 out of 40 motes in the network are selected as source nodes. Each source node periodically generates 40 information elements with an inter-element interval, denoted by Δ_r , uniformly distributed between 500ms and 3s. For *ONCR*, *MORE* and *CodeOR*, every consecutive eight information elements compose a batch. This pattern is to represent the light traffic scenario.
- *S20*: same as *S10* except that 20 nodes are selected as source nodes. This represents the heavy traffic scenario.

B. Measurement results

In what follows, we first present the measurement results for light traffic pattern *S10*, then we discuss the case of heavy traffic pattern *S20*. In the figures of this section, we present the means and their 95% confidence intervals for all metrics.

1) Light traffic For the light traffic pattern *S10*, Figures 3 - 5 show the delivery reliability, delivery cost and goodput of different protocols. We find that *ONCR* and *CTP* provide high data delivery reliabilities (both are close to 100%) while *MORE* and *CodeOR* can only delivery 78% and 85% of the data to the sink on average, respectively. Meanwhile, *ONCR* has a much lower delivery cost than *CTP*, i.e. a 26% reduction, in terms of average number of transmissions to deliver an information element but the delivery costs of *MORE* and *CodeOR* are around 400% and 300% of *CTP* respectively. The significant improvement in delivery cost also enables *ONCR* a higher data goodput than all other three protocols.

Experiment results of *ONCR* are consistent with Theorem 3 in that the transmission cost of minimal cost NC-based routing is upper-bounded by that of the shortest single path routing. Additionally, the design principles of *ONCR* in Section VI also contribute to these good characteristics by 1)loop avoidance mechanism in routing engine and 2)the rate control module that fills the gap between link-independence assumption and the possible link-correlation in real-world WSN. On the contrary, *MORE* and *CodeOR*, two representative NC-based OR protocols, perform poorly on sensor platforms.

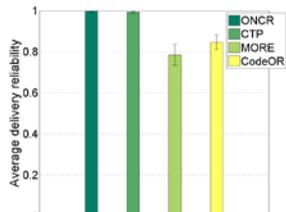


Fig. 3. Delivery reliability: light traffic

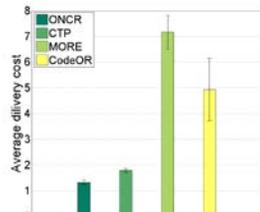


Fig. 4. Delivery cost: light traffic

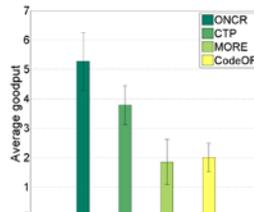


Fig. 5. Goodput: light traffic

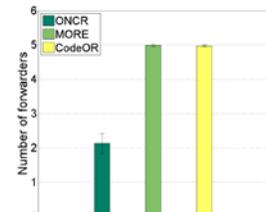


Fig. 6. Routing diversity: light traffic

There are several reasons why ONCR significantly outperforms MORE and CodeOR. First, having all forwarder candidates take part in the forwarding process could significantly increase the contention of network, resulting in drastic performance degradation. On the other hand, by only allowing candidates that can contribute in reducing the total transmission cost to join forwarder set, ONCR reduces channel contention and the corresponding packet transmission collision caused by opportunistic routing while still utilizing the wireless routing diversity towards improving the network performance. We verify this explanation by comparing the routing diversity of ONCR, MORE and CodeOR in Figure 8. It is shown that the average number of forwarders selected for each non-sink node in ONCR is around 2, but this number becomes 5 in MORE and CodeOR. This figure indicates that, though MORE and CodeOR are shown to achieve higher throughput than single path routing in wireless mesh networks [4][15], opportunistically selecting all candidates into forwarder set will impair the performance of NC-based routing, especially in resource-constrained wireless sensor networks.

Secondly, MORE relies heavily on a reliable end-to-end ACK channel to transmit feedback information from the destination to the whole network and so is CodeOR even if it uses hop-by-hop ACK as well. In wireless sensor networks, end-to-end ACK channel tends to be unreliable and it takes non-negligible time for all the nodes in the network to get an end-to-end ACK from the destination. In ONCR, we do not use any end-to-end ACK. Instead, our M-NSB ACK scheme enables a precise hop-by-hop feedback control for each sender to determine when to stop broadcasting, which help avoid redundant transmissions. Last but not least, the rate control module in ONCR also contributes to the performance improvement, as we discussed in Section VI, through reducing the probability of packet linearly dependence caused by link correlation. Rate control components in MORE and CodeOR could not fill this gap as ONCR does.

2) Heavy traffic To study the performance of ONCR in a more saturated network, we increase the number of sources to 20 to create a heavy traffic scenario S_{20} . Figures 7 - 9 present the delivery reliability, delivery cost and goodput of different protocols in S_{20} . With heavier traffic in the network, ONCR is still able to provide a 98% data delivery reliability as CTP does. Additionally, the reduction in delivery cost of ONCR compared to CTP has increased to 28%. With heavier data traffic in the network, the transmission cost of single path routing increases. On the contrary, the transmission cost of ONCR stays at a lower level in that it intelligently explores and optimally leverages the wireless routing diversity in the network. This observation is also consistent with Theorem 3 and the design principles of ONCR. And it again shows that NC-based routing protocols do not need to utilize full routing diversity to achieve the highest goodput.

Meanwhile, the performance of MORE and CodeOR degrades even more severely than CTP due to similar reasons

in the light traffic scenario. It is worthwhile to note that the goodput of CodeOR is even lower than MORE under S_{20} . This is because CodeOR tries to increase the concurrency of the network by allowing multiple flows from the same source to be injected in the network. However, with every node taking part in transmission opportunistically, this increase on concurrency would result in higher contention and poorer delivery performance in the network. Injecting too many flows in the network without considering the negative effects brought by opportunistic forwarding can be disastrous in a network with heavy traffic, as shown in our experiment results. We compare the routing diversity of all three NC-based protocols in Figure 10. It is shown that when traffic is heavier, ONCR tends to use even less routing diversity, i.e., less than 2 forwarders per node, while MORE and CodeOR still use as many forwarders as they are allowed to. This observation demonstrates, from another perspective, the importance and necessity to optimally select forwarder sets for NC-based routing protocols.

VIII. RELATED WORK

In-network processing (INP) has been well studied in wireless and sensor networks. Many INP methods has been proposed to improve network performance by reducing data traffic [14]. These methods are divided into application-dependent INP, e.g., data aggregation and fusion, and application-independent INP, e.g., packet packing [21], compressive sensing [12] and network coding [2]. Network coding was first proposed for wired networks [2]. By mixing packets at intermediate nodes, the bandwidth can be saved and network throughput can be significantly improved. In the past decade, network coding has been extensively studied not only to improve throughput, but also to provide network protection [22][10]. It was later extended to wireless networks due to the broadcast nature of wireless communication. Ho *et al.* [9] proposed random network coding (RNC) and proved that it achieves the same throughput performance as deterministic linear coding. RNC has been widely used in wireless systems, including mesh networks [8][23], mobile network [18] and sensor networks [19][7].

One important application of random network coding is to be incorporated into routing protocols to improve the throughput of wireless networks. Gkantsidis *et al.* [5] theoretically analyzed the throughput gain brought by random network coding in multipath routing protocols. Chachulski *et al.* [4] proposed MORE, the first NC-based opportunistic routing protocol in wireless networks. Integrating intra-flow random network coding with a state-of-the-art opportunistic routing (OR) protocol [3], MORE fully utilizes the routing diversity in wireless environment by allowing every node to broadcast randomly coded packets in the forwarding process. The randomness brought by RNC eliminates the need of coordination between nodes in the same forwarder candidate set, making NC-based OR become more favorable in wireless mesh networks than pure opportunistic routing. Quite a few



Fig. 7. Delivery reliability: heavy traffic

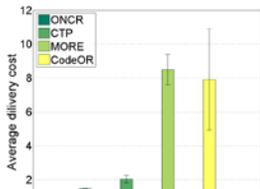


Fig. 8. Delivery cost: heavy traffic

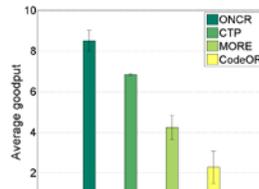


Fig. 9. Goodput: heavy traffic

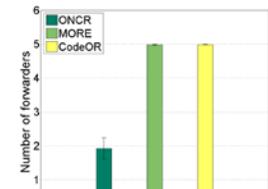


Fig. 10. Routing diversity: heavy traffic

NC-based OR protocols have been proposed using MORE as a prototype [11][15][13]. Katti *et al.* [11] propose MIXIT, which performs physical layer coding on a symbol-level to enhance the throughput and reliability of mesh networks. Lin *et al.* [15] added hop-by-hop ACK and sliding window to MORE in the design of CodeOR, an NC-based OR protocol aiming to increase the concurrency of network. Koutsonikolas *et al.* [13] designed a cumulative coded feedback scheme to address collective-space problem and false-positive problem in feedback control of NC-based OR. Based on this scheme, the authors designed CCACK, another NC-based OR protocol. MIXIT, CodeOR and CCACK are shown to provide significant improvement on throughput in mesh networks.

Our work is closely related to [4][11][15][13]. However, these studies do not consider forwarder set selection problem and instead *allow every forwarder candidate to join forwarder set*. The increase of contention brought by opportunistic routing and the corresponding increase in transmission cost are neglected in all these studies. MIXIT adopts a credit assignment algorithm sharing the similar idea of our Algorithm 1 for the **D-EST-NC** problem. But this credit assignment only aims to reduce the queuing delay at each node and still utilizes all the routing diversity without minimizing the transmission cost. Our work, on the contrary, shows that the opportunistic routing principle used in these protocols would lead to drastic performance degradation in resource-constrained wireless networks. We focus on selecting the optimal forwarder set for each node in the network such that the total number of packet transmissions is minimized. Our analytical and experimental results show that the performance of NC-based opportunistic routing is poor in resource-constrained sensor networks, and demonstrate the importance and necessity of minimal cost forwarder set selection for wireless NC-based routing protocol.

IX. CONCLUDING REMARKS

We studied the problems of estimating and minimizing the cost of NC-based routing. In particular, we propose the first analytical framework for estimating the expected total cost of NC-based routing. This framework is applicable to arbitrary topologies with arbitrary forwarder set compositions. We design a greedy forwarder set selection algorithm for NC-based routing and proved its optimality. We have also analytically shown the advantages of minimal cost NC-based routing over shortest single path routing. Based on these findings, we design ONCR, a minimal cost NC-based routing protocol. In ONCR, the M-NSB ACK scheme enables precise feedback control of coded message forwarding, and the rate control component reduces the potential linear dependency between coded packets due to link correlation. Experimental results on the NetEye sensor testbed show that ONCR provides a close to 100% delivery reliability while incurring a transmission cost of only about 72-75% of the best existing protocol. Meanwhile, ONCR achieves about twice the goodput of MORE and CodeOR in resource-constrained sensor networks. This paper reveals that opportunistic forwarder set selection could cause

disastrous performance degradation for NC-based routing in resource-constrained wireless networks. Our findings shed light on how to optimally utilize routing diversity in wireless communication.

X. ACKNOWLEDGMENT

The authors' work are partially supported by NSF awards CNS-1136007, CNS-1054634, CNS-1239108, CNS-1218718, CNS-1423102, GENI-1890, GENI-1633 and IIS-1231680, NSERC Collaborative Research and Development Grant (CRDJP 418713) and CFI Leaders Opportunity Fund 23090.

REFERENCES

- [1] NetEye testbed. <http://neteye.cs.wayne.edu/neteye/home.php>.
- [2] R. Ahlswede, N. Cai, S.-Y. Li, and R. Yeung. Network information flow. *Information Theory, IEEE Transactions on*, 2000.
- [3] S. Biswas and R. Morris. ExOR: Opportunistic multi-hop routing for wireless networks. In *ACM SIGCOMM*, 2005.
- [4] S. Chachulski, M. Jennings, S. Katti, and D. Katabi. Trading structure for randomness in wireless opportunistic routing. In *SIGCOMM 2007*.
- [5] C. Gkantsidis, W. Hu, P. Key, B. Radunovic, and P. Rodriguez. Multipath code casting for wireless mesh networks. In *CoNEXT 07*.
- [6] O. Gnawali, R. Fonseca, K. Jamieson, D. Moss, and P. Levis. Collection tree protocol. In *ACM SenSys*, 2009.
- [7] Z. Guo, B. Wang, P. Xie, W. Zeng, and J.-H. Cui. Efficient error recovery with network coding in underwater sensor networks. *Ad Hoc Networks*.
- [8] M. K. Han, A. Bhartia, L. Qiu, and E. Rozner. O3: Optimized overlay-based opportunistic routing. In *MobiHoc '11*.
- [9] T. Ho, R. Koetter, M. Medard, D. Karger, and M. Effros. The benefits of coding over routing in a randomized setting. In *ISIT*, 2003.
- [10] A. E. Kamal and O. Al-Kofahi. Efficient and agile 1+ n protection. *Communications, IEEE Transactions on*, 59(1):169–180, 2011.
- [11] S. Katti, D. Katabi, H. Balakrishnan, and M. Medard. Symbol-level network coding for wireless mesh networks. In *ACM SIGCOMM 08'*.
- [12] L. Kong, M. Xia, X. Liu, G. Chen, Y. Gu, M. Wu, and X. Liu. Data loss and reconstruction in wireless sensor networks. *IEEE TPDS*, 2014.
- [13] D. Koutsonikolas, C.-C. Wang, and Y. Hu. Ccack: Efficient network coding based opportunistic routing through cumulative coded acknowledgments. In *INFOCOM, 2010*.
- [14] R. Kumar, B. Agarwalla, J. Shin, P. Hutto, A. Paul, and U. Ramachandran. DFuse: A framework for distributed data fusion. In *SenSys 03'*.
- [15] Y. Lin, B. Li, and B. Liang. Codeor: Opportunistic routing in wireless mesh networks with segmented network coding. In *ICNP 2008*.
- [16] J.-S. Park, M. Gerla, D. Lun, Y. Yi, and M. Medard. Codecast: a network coding based ad hoc multicast protocol. *Wireless Communications*, 2006.
- [17] C. Reis, R. Mahajan, M. Rodrig, D. Wetherall, and J. Zahorjan. Measurement-based models of delivery and interference in static wireless networks. *ACM SIGCOMM*, 36(4):51–62, 2006.
- [18] B. Tang, B. Ye, S. Guo, S. Lu, and D. Wu. Order-optimal information dissemination in manets via network coding. *IEEE TPDS*, July 2014.
- [19] S. Wang, G. Tan, Y. Liu, H. Jiang, and T. He. Coding opportunity aware backbone metrics for broadcast in wireless networks. In *INFOCOM '13*.
- [20] Q. Xiang. In-network processing for mission-critical wireless networked sensing and control: A real-time, efficiency, and resiliency perspective. Dissertation, Wayne State University.
- [21] Q. Xiang, H. Zhang, J. Xu, X. Liu, and L. Rittle. When in-network processing meets time: Complexity and effects of joint optimization in wireless sensor networks. *IEEE TMC*, 2011.
- [22] H. Yao, S. Jaggi, and M. Chen. Network coding tomography for network failures. In *INFOCOM, 2010 Proceedings IEEE*, pages 1–5. IEEE, 2010.
- [23] X. Zhang and B. Li. Optimized multipath network coding in lossy wireless networks. *IEEE JSAC*, 27(5):622–634, 2009.