EENCR: An Energy-efficient Network Coding based Routing Protocol

May 8, 2014

1 Preliminary

Ahlswede *et al.* [2] first proposed the network coding technique. The authors showed that the use of network coding can effectively increase throughput in wired networks. Since then, different network coding strategies have been studied, e.g., linear network coding [2], non-linear network coding [17] and random network coding [11] [12]. Ho et al. [11] proved that the use of random network coding can achieve the theoretical maximal throughput in wireless networks. And Eryilmaz *et al.* [7] showed that network coding can reduce transmission latency, therefore can increase the throughput in multicast traffic flow.

Recently, Chachulski *et al.* [4] proposed MORE, the first protocol to integrate random network coding with opportunistic routing for unicast flow in wireless mesh networks. Experiment results show that MORE yields a higher network throughput than ExOR [3] which only uses opportunistic routing. Based on the framework of [4], [19] [14] further improve the network throughput by introducing different ACK and rate control schemes. To the best of our knowledge, however, there has been no systemic study on spatial and energy consumption control on network-coding-based (NC-based) routing, which is of great importance in power-constrained distributed systems, e.g., wireless sensor networks.

In this work, we study the open problem of minimal cost NC-based routing in wireless networks. Our main contributions are as follows.

- We propose an effective load based approach to measure the expected number of transmissions of NC-based transmission for arbitrary topologies. This is the first mathematical framework to compute the transmission cost of NC-based routing.

- We propose a polynomial greedy algorithm to compute the minimal transmission cost and the corresponding routing braid for NC-based routing. We prove the optimality of this algorithm and an upper bound of transmission cost for the optimal NC-based routing braid, which is equal to the cost of shortest single path routing.

- Based on the algorithm we proposed, we design and implement EENCR, an energy-efficient NC-based routing protocol, for resource-constrained sensor platforms. In EENCR, we incorporate the 4-bit link estimator of CTP [9], realize a light-weight distributed implementation for our greedy forwarder set selection algorithm in the rouging engine, and design a modified null-spaced-based (M-NSB) coded feedback scheme and a corresponding rate control component. Compared to CTP, EENCR introduces zero additional communication cost but yields an optimal routing braid with lower cost than the shortest single path routing.

- We evaluate the performance of EENCR on the NetEye testbed by comparing it with CTP [9], MORE [4] and CodeOR [19]. Experiment results show that EENCR achieves a close to 100% reliability with a large transmission cost reduction of CTP, i.e., 25 - 28%. And EENCR further improves the goodput of NC-based routing protocol by adaptively selecting the forwarders instead of utilizing the whole forwarder candidate set.

The remaining of this chapter is organized as follows. We first introduce the system settings and problem definition. We then propose the effective-load-based framework to compute the transmission cost of NC-based routing. Based on this framework, we design a polynomial-time greedy algorithm that can compute the optimal routing braid for arbitrary topologies. Next we present EENCR, which includes a distributed implementation of our greedy algorithm. We evaluate the performance of EENCR under different topologies on the NetEye testbed. Before we conclude this chapter, we discuss related work in the field of network coding.

2 System settings and problem definition

In this section, we first present the system settings we used in this study. Next we explain why we choose intra-flow network coding in designing efficient routing protocol for mission-critical WCPS. Based on the system model, we formally define the problems of transmission cost computation and optimization for NC-based routing.

2.1 System settings

In this study, we model a wireless network as a directed graph $G = (V, E)$ with node S as the source and T as the destination. For each node $i \in G$, we use U_i and D_i to denote the set of senders and receivers of i, respectively. And we denote the forwarder set of i as $FS_i \subset D_i$. For each link $i \to j \in E$, we denote ETX_{ij}^x as its expected number of transmission to deliver a packet with length x and $P_{ij}^x = \frac{1}{ETX_{ij}}$ as the corresponding link reliability. Since network coding will not change the packet length during the transmission, we use ETX_{ij} and P_{ij} for simplicity. Then we define $C_{iT}(x)$ as the transmission cost of delivering x linear independent packets from i to T, and $C_{iD_i}(x)$ as the expected number of broadcasts of node i when nodes in D_i collectively receive x linear independent coded packets from i. Assuming S needs to deliver K packets as a batch to T, we define K_i^j as the number of linear independent packets node i received from node j.

2.2 Problem definition

We define the minimal cost NC-based routing problem as follows:

Problem $\mathbf{Q_0}$ Given a directed graph $G = (V, E)$ with one source S and one destination T, find the optimal total transmission cost and the corresponding FS_i for each node i to deliver K packets using intra-flow random network coding from S to T.

To the best of our knowledge, however, there has been no study on how to measure the transmission cost of intra-flow network coding, letting alone the optimal transmission cost. Therefore, we need to first find a way to measure the transmission cost of NC-based routing before we can solve $\mathbf{Q_0}$. Therefore, we define the following problem:

Problem $\mathbf{Q_1}$ the same as $\mathbf{Q_0}$ except that $FS_i = D_i$ for each node i.

The solution to problem $\mathbf{Q_1}$ can provide a mathematical framework to compute the expected transmission cost of NC-based routing. Not only will this framework provide a tool for our solution to problem Q_0, but also it will shed lights towards energy-efficiency study of NC-based transmission in future study. In the following sections, we will propose optimal polynomial-time algorithms for both problem $\mathbf{Q_1}$ and $\mathbf{Q_0}$.

3 Cost optimization for NC-based routing

In this section, we first propose an effective load assignment algorithm to solve problem $\mathbf{Q_1}$. The key idea of this algorithm is to compute the number of encoded packets each intermediate node should forward and the corresponding cost. Based on this approach, we then design a distributed polynomial-time algorithm to optimally solve $\mathbf{Q_0}$. For each node i in $\mathbf{Q_0}$, we choose the forwarder set FS_i out of D_i using a greedy algorithm based on the transmission cost from each node in D_i to T We prove this algorithm's optimality and show that the optimal transmission cost of NC-based routing has an upper bound that equals to the transmission cost of the shortest path routing.

3.1 Effective load based assignment algorithm for $\mathbf{Q_1}$

In NC-based opportunistic routing protocols, such as MORE [4], the network throughput is significantly improved compared with single path routing. However, the transmission cost of these protocols are not carefully controlled and it may be higher than the cost of single path routing since every intermediate node will forwards re-encoded packets to its own forwarder candidate set. To precisely measure and control the transmission cost while still fully utilizing the benefit of network coding on throughput, we propose a concept called *effective load*.

Definition 1 *For a node j in the forwarder candidate set FCS_i, the* ***effective load*** *L_j is defined as the number of linear independent packets that are received by j but not by any of the other nodes in FCS_i that has lower transmission cost to the destination.*

To demonstrate this concept, we first look at the following example in Figure 1. In this example, the source node S has $K = 3$ packets that needs delivering to T and $C_{AT} < C_{BT}$. Therefore, node A has a higher priority than B in FCS_S. When S stops broadcasting, the coding vectors of packets received by node A are $\{1, 2, 3\}$ and $\{1, 1, 1\}$ and the vectors at node

B are $\{2, 3, 5\}$ and $\{1, 1, 1\}$. Since node A has a lower transmission cost to T than B, node A has an effective load $L_A = 2$. Node B only has an effective load $L_B = 1$ because the packet with coding coefficient $\{1, 1, 1\}$ is also received by A. If both node A and B forward up to their effective load of re-encoded packets to T, T will receive 3 linear independent packets, which is just enough to decode the whole batch. In the meantime, there will be no unnecessary re-encoding forwarding operations from FSC_S to T.

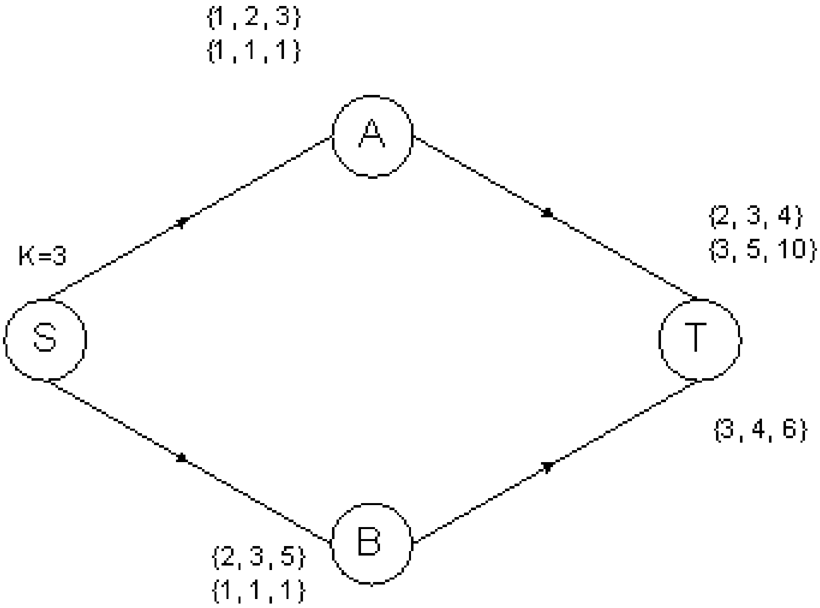

Figure 1: An illustrating example of NC-based routing

Based on the concept of effective load, we then propose a framework to compute the transmission cost of NC-based routing based on different effective load between nodes within the same forwarder candidate set, i.e., given a node i, each node $j \in D_i$ will forward L_j linear independent packets to the destination

To better illustrate how to use the effective load approach to compute the transmission cost of NC-based routing, we first study the following example in Figure 2.

In this diamond topology, we define that $P_2 \geq P_4 \geq P_6$. The whole transmission process can be divided into two steps. The first step is node S broadcasting to $D_S = \{A, B, C\}$ and the second step is nodes in D_S forwarding re-encoded packets to T. In the first step, we treat node A, B, C as one single virtual node V_{D_S}. The link reliability of link $S \rightarrow V_{D_S}$ is then expressed

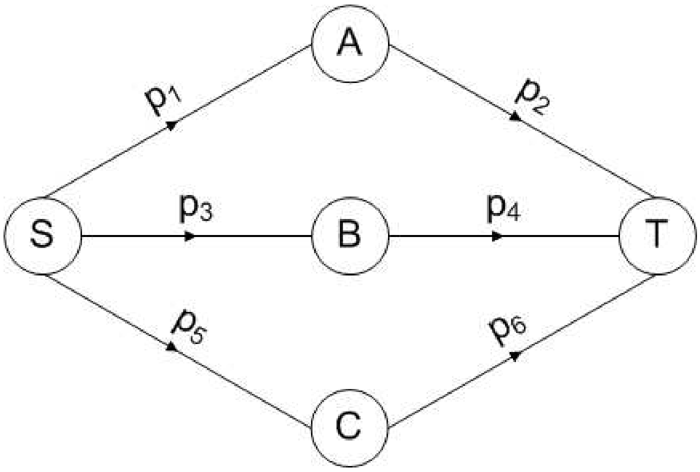

Figure 2: Example topology

as $P_{SV_{D_S}} = 1-(1-P_1)(1-P_3)(1-P_5)$. Therefore, the transmission cost for the first step is

$$C_{SD_S}(K) = \frac{K}{P_{SV_{D_S}}} = \frac{K}{1-(1-P_1)(1-P_3)(1-P_5)} \tag{1}$$

In the second step, since we have $P_2 \geq P_4 \geq P_6$, we want path $A \rightarrow T$ to forward as many packets as it is capable of and path $C \rightarrow T$ to forward as least packets as needed. To compute the effective load for nodes inD_S, we first compute K_i, the expected number of linear independent packets received by each node from S in the first step.

$$\begin{cases} K_A^S = & \dfrac{KP_1}{1-(1-P_1)(1-P_3)(1-P_5)} \\ K_B^S = & \dfrac{KP_3}{1-(1-P_1)(1-P_3)(1-P_5)} \\ K_C^S = & \dfrac{KP_5}{1-(1-P_1)(1-P_3)(1-P_5)} \end{cases} \tag{2}$$

Using L_i to denote the number of linear independent packets node i needs to forward to T, it is easy to see that $L_A = K_A^S$. However, we cannot simply calculate L_B as $min(K - L_A, K_B^S)$ because node B and A may receive some same packets, resulting in less entropy held by B. Instead, we need to compute $K_B^{S'}$, the expected number of linear independent packets that are received by node B but not A.

$$K_B^{S'} = K\frac{K_B^S}{K}(1-P_1) = K_B^S(1-P_1) \tag{3}$$

The detailed deduction to compute $K_B^{S'}$ is to solve an easy probability theory problem and is hence omitted. It is easy to see that $K_B^{S'} < K - L_A$, thus we have $L_B = K_B S'$. Similarly, we have $L_C = K_C^{S'} = K_C^S(1-P_1)(1-P_3)$ and we can verify that $L_A + L_B + L_C = K$. Combining these intermediate results, we have the total transmission cost computed as:

$$\begin{aligned} C_S(K) &= C_{SD_S}(K) + C_{AT}(L_A) \\ &\quad + C_{BT}(L_B) + C_{CT}(L_C) \\ &= \frac{K}{1-(1-P_1)(1-P_3)(1-P_5)} \\ &\quad + \frac{L_A}{P_2} + \frac{L_B}{P_4} + \frac{L_C}{P_6} \\ &= \frac{K}{1-(1-P_1)(1-P_3)(1-P_5)} \\ &\quad [1 + \frac{P_1}{P_2} + \frac{P_3(1-P_1)}{P_4} + \frac{P_5(1-P_1)(1-P_3)}{P_6}] \end{aligned} \tag{4}$$

Through this example, we demonstrate how to compute the transmission cost of NC-based routing. The basic idea is to first compute the broadcast cost treating all the nodes in the forwarder candidate set as one single virtual node, and then compute the effective load, i.e., the number of re-encoded packets needs to be forwarded at each node in the forwarder candidate set based on an non-decreasing order of their cost to the destination.

Although the topology in Figure 2 consists of only node-disjoint paths from the source to the destination, we can generalize this approach to recursively compute the cost of NC-based routing in arbitrary topologies. We formally present this computing process as Algorithm 1. Basically, each node i runs Algorithm 1 to compute its transmission cost to the destination if every node in D_i has its transmission cost computed and updated. Node i then sends its own cost information to its sender(s). The sender(s) then run this algorithm again to compute their transmission cost to the destination. By the end of this backwards recursive process, the source node S will be able to compute its transmission cost to the destination based on the transmission cost of nodes in D_S. Note that the complexity of Algorithm 1 is $O(|V| \lg |V|)$, which makes it suitable for power-constrained computation platforms, e.g., the Telosb sensor platform.

Algorithm 1 Compute the transmission cost of NC-based routing for the current node S with M forwarder candidates

```
Input: current node $S$, $D_S = \{A_1, A_2, \ldots, A_M\}$
Output: $C_S(1)$: the expected number of transmissions to deliver 1 packet from $S$ to $T$
Sort nodes in $D_S$ by a non-descending order of $C_{A_i}(1)$, where $i = 1, 2, \ldots, M$.
Sorted nodes are labeled as $\{A'_1, A'_2, \ldots, A'_M\}$
$C_{SD_S}(1) = \frac{1}{1-\prod_{i=1}^{M}(1-P_{SA'_i})}$
$L_{A'_1} = C_{SD_S}(1)P_{SA'_1}$
$F = 1 - P_{SA'_1}$
for $i \to 2, 3, \ldots, M$ do
    $L_{A'_i} = C_{SD_S}(1)P_{SA'_i}F$
    $C_{A'_i}(L_{A'_i}) = L_{A'_i}C_{A'_i}(1)$
    $F = F(1 - P_{SA'_i})$
end for
$C_S(1) = C_{SD_S}(1) + \sum_{i=1}^{M} C_{A'_i}(L_{A'_i})$
```

As described above, the major principle we use here is to always assign more traffic load to forwarders with lower cost, which implies that we should apply different utilization of forwarders in the DAG to minimize the transmission cost instead of fully utilizing every possible path in the network. This observation provides two insights: 1)it shows that we do not need full network coding redundancy in the network to perform regular data transmission, which would cause higher transmission cost and contention; 2) extra redundancy may be used to provide proactive protection to mission-critical networks against single node failures. These two insights lead us to the solution to problem $\mathbf{Q_0}$ in this chapter and problem $\mathbf{Q}$ in Chapter 4.

3.2 Optimal NC-based transmission cost algorithm

In the last section, we proposed a distributed algorithm executed by each node to compute the transmission cost of NC-based routing from a given source to the destination. However, there still lacks a precise control on transmission cost in NC-based routing, making NC-based transmission energy-inefficient. This energy-inefficiency is especially severe in dense networks where each node has many forwarder candidates.

In MORE-based protocols, forwarder candidates with low expected effective load are usually not allowed to forward the flow to reduce the contention in the network, which can reduce the transmission cost sometimes. However, this reduction is not guaranteed and sometimes it may even increase the transmission cost. Based on the observations we had a priori, we design a distributed greedy algorithm, Algorithm 2. The basic idea of Algorithm 2 is as follows. For an input node S, we 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 with the lowest transmission cost from the sorted D_S, add it to the forwarder set FS_S and compute the total transmission cost using Algorithm 1. If the 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 loop until either of the following two conditions is satisfied:

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

Each non-destination node executes this algorithm to determine the minimal transmission cost from itself to the destination T and the corresponding forwarder candidate set. Upon the convergence of the whole network, we will get the solution to problem $\mathbf{Q_0}$.

The complexity of this algorithm is $O(|V|^2 \lg |V|)$. In NC-based routing, the size of FC_S can also be one and in this case the routing braid is the equivalent to the shortest single path. Next we show readers the optimality of this algorithm by proving the following theorem:

Theorem 1 *Given a node S and its forwarder candidate set $D_S = \{A_1, A_2, \ldots, A_M\}$, Algorithm 2 yields the minimal transmission cost to the destination node of NC-based routing and the corresponding forwarder set.*

Proof We prove the correctness of this theorem by contradiction. Given a node S and its forwarder candidate set D_S, we denote the minimal transmission cost as C^* and the corresponding transmission forwarder set is FS^*_S with a cardinality of k. We sort nodes in FS^*_S in

Algorithm 2 Compute the minimal transmission cost of NC-based routing and the corresponding FCS for the input node S with M forwarders

```
Input: node S, D_S = {A_1, A_2, ..., A_M}, FS_S = ∅
Output: C*_S(1): the minimal transmission cost to deliver 1 packet from S to T
Sort nodes in D_S by a non-descending order of C_{A_i}(1), where i = 1, 2, ..., M.
Sorted nodes are labeled as {A'_1, A'_2, ..., A'_M}
FS_S = {A'_1}
C*_S(1) = 1/P_{SA'_1} + C_{A'_1}(1)
for i → 2, 3, ..., M do
   Run Algorithm 1 with input S and D_S = {A'_1, ..., A'_i}
   Get the result as C^{new}_S(1)
   if C^{new}_S(1) > C_S(1) then
      break
   else
      FS_S = FS_S ∪ A'_i
      C*_S(1) = C^{new}_S(1)
   end if
end for
```

non-descending order of their transmission cost to the destination and denote them as $FS_S^* = \{A_1^*, A_2^*, \ldots, A_k^*\}$ where $C_{A_1^*} \leq C_{A_2^*} \leq, \ldots, \leq C_{A_k^*}$.

If this theorem is not correct, then there exists at least one node A_x having $C_{A_x} \leq C_{A_i^*}$ for some integer $i \in [1, k]$. Without loss of generality, we assume that $C_{A_{k-1}^*} \leq C_{A_x} \leq C_{A_k^*}$. We will have a contradiction when we can find a forwarder set FS_S^{**} that has a lower transmission cost C^{**} than C^*. To find this contradiction, we study the following forwarder sets:

$$\begin{aligned} FS_S^* &= \{A_1^*, A_2^*, \ldots, A_k^*\} \\ FS_S^1 &= FS_S^* - \{A_k^*\} \\ FS_S^2 &= FS_S^* \cup \{A_x\} \end{aligned} \tag{5}$$

For each forwarder set, we compute the transmission cost for these forwarder sets using Algorithm 1. The transmission cost of FS_S^* is expressed as:

$$C^* = \frac{1 + \sum_{i=1}^{k}[C_{A_i^*} P_{SA_i^*} \prod_{j=1}^{i-1}(1 - P_{SA_j^*})]}{1 - \prod_{i=1}^{k}(1 - P_{SA_i^*})} \tag{6}$$

Compared with FS_S^*, FS_S^1 does not have node A_k^*, therefore cost C_1 is expressed as:

$$C_1 = \frac{1 + \sum_{i=1}^{k-1}[C_{A_i^*} P_{SA_i^*} \prod_{j=1}^{i-1}(1 - P_{SA_j^*})]}{1 - \prod_{i=1}^{k-1}(1 - P_{SA_i^*})} \tag{7}$$

On the other hand, FS_S^2 consists of both FS_S^* and node A_x. Since A_x has a lower transmission cost than node A_k^*, we compute C_2 as:

$$\begin{aligned} C_2 = & \frac{1}{1 - (1 - P_{SA_x}) \prod_{i=1}^{k}(1 - P_{SA_i^*})} \\ & \cdot \{1 + \sum_{i=1}^{k-1}[C_{A_i^*} P_{SA_i^*} \prod_{j=1}^{i-1}(1 - P_{SA_j^*})] + C_{A_x} P_{SA_x} \prod_{i=1}^{k-1}(1 - P_{SA_i^*}) \\ & + C_{A_k^*} P_{SA_k^*}(1 - P_{SA_x}) \prod_{i=1}^{k-1}(1 - P_{SA_i^*})\} \end{aligned} \tag{8}$$

Based on our assumption, C^*, C_1 and C_2 have the following relations:

$$\begin{aligned} C^* - C_1 & \leq 0 \\ C^* - C_2 & \leq 0 \end{aligned} \tag{9}$$

The basic idea next is to prove that $C^* - C_2 \geq 0$ when $C^* - C_1 \leq 0$, which leads to a contradiction. To accomplish this goal, we conduct some mathematical transformation of two inequities above. The first inequity is between C^* and C_1. Starting from the fact that $\frac{C^*}{C_1} \leq 1$, we have the following useful result:

$$
\begin{aligned}
& \tfrac{C^*}{C_1} = \frac{1-\prod_{i=1}^{k-1}(1-P_{SA_i^*})}{1-\prod_{i=1}^{k}(1-P_{SA_i^*})} \cdot \frac{1+\sum_{i=1}^{k}[C_{A_i^*}P_{SA_i^*}\prod_{j=1}^{i-1}(1-P_{SA_j^*})]}{1+\sum_{i=1}^{k-1}[C_{A_i^*}P_{SA_i^*}\prod_{j=1}^{i-1}(1-P_{SA_j^*})]} \leq 1 \\
\Leftrightarrow \quad & [1-\textstyle\prod_{i=1}^{k-1}(1-P_{SA_i^*})] \cdot \{1+\sum_{i=1}^{k}[C_{A_i^*}P_{SA_i^*}\prod_{j=1}^{i-1}(1-P_{SA_j^*})]\} \\
& -[1-\textstyle\prod_{i=1}^{k}(1-P_{SA_i^*})] \cdot \{1+\sum_{i=1}^{k-1}[C_{A_i^*}P_{SA_i^*}\prod_{j=1}^{i-1}(1-P_{SA_j^*})]\} \leq 0 \\
\Leftrightarrow \quad & \{1+\textstyle\sum_{i=1}^{k-1}[C_{A_i^*}P_{SA_i^*}\prod_{j=1}^{i-1}(1-P_{SA_j^*})]\} \cdot \{\prod_{i=1}^{k}(1-P_{SA_i^*}) - \prod_{i=1}^{k-1}(1-P_{SA_i^*})\} \\
& +[1-\textstyle\prod_{i=1}^{k-1}(1-P_{SA_i^*})]P_{SA_k^*}C_{A_k^*}\prod_{i=1}^{k-1}(1-P_{SA_i^*}) \leq 0 \\
\Leftrightarrow \quad & \{1+\textstyle\sum_{i=1}^{k-1}[C_{A_i^*}P_{SA_i^*}\prod_{j=1}^{i-1}(1-P_{SA_j^*})]\} \cdot (-P_{SA_k^*}) \\
& +[1-\textstyle\prod_{i=1}^{k-1}(1-P_{SA_i^*})]P_{SA_k^*}C_{A_k^*} \leq 0 \\
\Leftrightarrow \quad & \\
& 1+\textstyle\sum_{i=1}^{k-1}[C_{A_i^*}P_{SA_i^*}\prod_{j=1}^{i-1}(1-P_{SA_j^*})] \geq [1-\prod_{i=1}^{k-1}(1-P_{SA_i^*})]C_{A_k^*}
\end{aligned}
\tag{10}
$$

And the second result is between C^* and C_2 and this time we directly expand the difference:

$$
\begin{aligned}
C^* - C_2 \quad = \quad & \frac{1+\sum_{i=1}^{k}[C_{A_i^*}P_{SA_i^*}\prod_{j=1}^{i-1}(1-P_{SA_j^*})]}{1-\prod_{i=1}^{k}(1-P_{SA_i^*})} \\
& -\frac{1}{1-(1-P_{SA_x})\prod_{i=1}^{k}(1-P_{SA_i^*})} \\
& \cdot\{1+\textstyle\sum_{i=1}^{k-1}[C_{A_i^*}P_{SA_i^*}\prod_{j=1}^{i-1}(1-P_{SA_j^*})] + C_{A_x}P_{SA_x}\prod_{i=1}^{k-1}(1-P_{SA_i^*}) \\
& +C_{A_k^*}P_{SA_k^*}(1-P_{SA_x})\textstyle\prod_{i=1}^{k-1}(1-P_{SA_i^*})\} \\
= \quad & \{1+\textstyle\sum_{i=1}^{k-1}[C_{A_i^*}P_{SA_i^*}\prod_{j=1}^{i-1}(1-P_{SA_j^*})]\} \\
& \cdot\{\frac{1}{1-\prod_{i=1}^{k}(1-P_{SA_i^*})} - \frac{1}{1-(1-P_{SA_x})\prod_{i=1}^{k}(1-P_{SA_i^*})}\} \\
& +\textstyle\prod_{i=1}^{k-1}(1-P_{SA_i^*}) \\
& \cdot\{P_{SA_k^*}C_{A_k^*}[\frac{1}{1-\prod_{i=1}^{k}(1-P_{SA_i^*})} - \frac{1-P_{SA_x}}{1-(1-P_{SA_x})\prod_{i=1}^{k}(1-P_{SA_i^*})}] \\
& -P_{SA_x}C_{A_x} \cdot \frac{1}{1-(1-P_{SA_x})\prod_{i=1}^{k}(1-P_{SA_i^*})}\}
\end{aligned}
\tag{11}
$$

Using some simple technique, we further transform the right-hand of Equation 11 and have the following result:

$$\begin{aligned}
C^* - C_2 &= \{1 + \textstyle\sum_{i=1}^{k-1}[C_{A_i^*} P_{SA_i^*} \prod_{j=1}^{i-1}(1 - P_{SA_j^*})]\} \\
&\quad \cdot \frac{P_{SA_x} \prod_{i=1}^{k}(1 - P_{SA_i^*})}{[1 - \prod_{i=1}^{k}(1 - P_{SA_i^*})][1 - (1 - P_{SA_x}) \prod_{i=1}^{k}(1 - P_{SA_i^*})]} \\
&\quad + \frac{P_{SA_x} \prod_{i=1}^{k-1}(1 - P_{SA_i^*})}{1 - (1 - P_{SA_x}) \prod_{i=1}^{k}(1 - P_{SA_i^*})} \cdot [\frac{P_{SA_k^*} C_{A_k^*}}{1 - \prod_{i=1}^{k}(1 - P_{SA_i^*})} - C_{A_x}] \\
&= \frac{P_{SA_x} \prod_{i=1}^{k-1}(1 - P_{SA_i^*})}{[1 - \prod_{i=1}^{k}(1 - P_{SA_i^*})][1 - (1 - P_{SA_x}) \prod_{i=1}^{k}(1 - P_{SA_i^*})]} \\
&\quad \cdot \{\{1 + \textstyle\sum_{i=1}^{k-1}[C_{A_i^*} P_{SA_i^*} \prod_{j=1}^{i-1}(1 - P_{SA_j^*})]\}(1 - P_{SA_k^*}) \\
&\quad + P_{SA_k^*} C_{A_k^*} - [1 - \textstyle\prod_{i=1}^{k}(1 - P_{SA_i^*})] C_{A_x}\} \\
&= \frac{P_{SA_x} \prod_{i=1}^{k-1}(1 - P_{SA_i^*})}{[1 - \prod_{i=1}^{k}(1 - P_{SA_i^*})][1 - (1 - P_{SA_x}) \prod_{i=1}^{k}(1 - P_{SA_i^*})]} \\
&\quad \cdot \{\{1 + \textstyle\sum_{i=1}^{k-1}[C_{A_i^*} P_{SA_i^*} \prod_{j=1}^{i-1}(1 - P_{SA_j^*})]\}(1 - P_{SA_k^*}) \\
&\quad + P_{SA_k^*} C_{A_k^*} - P_{SA_k^*} C_{A_x} - [1 - P_{SA_k^*} - \textstyle\prod_{i=1}^{k}(1 - P_{SA_i^*})] C_{A_x} \\
&= \frac{P_{SA_x} \prod_{i=1}^{k-1}(1 - P_{SA_i^*})}{[1 - \prod_{i=1}^{k}(1 - P_{SA_i^*})][1 - (1 - P_{SA_x}) \prod_{i=1}^{k}(1 - P_{SA_i^*})]} \\
&\quad \{(1 - P_{SA_k^*})\{1 + \textstyle\sum_{i=1}^{k-1}[C_{A_i^*} P_{SA_i^*} \prod_{j=1}^{i-1}(1 - P_{SA_j^*})] \\
&\quad - [1 - \textstyle\prod_{i=1}^{k}(1 - P_{SA_i^*})] C_{A_x}\} \\
&\quad + P_{SA_k^*} C_{A_k^*} - P_{SA_k^*} C_{A_x}\}
\end{aligned} \tag{12}$$

Using the result of Inequality 10 and the fact that $C_{A_k^*} > C_{A_x}$, we can find that the right hand side of Equation 12 is greater than 0, which means $C^* > C_2$ and shows the existence of a contradiction. And we note that using the above mathematical deduction framework, a contradiction can be found for any number of is where $i \in [1, k]$ and $C_{A_i^*} > C_{A_x}$. Therefore, we proved that we can find the minimal transmission cost of S to the destination by adding forwarder candidates with lower transmission cost to the destination into the forwarder set until adding more candidates will increase the C_S. By now, we complete our proof on the optimality of Algorithm 2 in computing the optimal NC-based routing topology.

3.3 A theoretical comparison with other routing protocols

In the previous section, we proposed an optimal greedy algorithm that computes the minimal transmission cost of NC-based routing. Different from the heuristic control of spatial diversity in other MORE-based network coding opportunistic routing protocols, this algorithm intelligently explores the routing diversity in wireless transmission and only adds routes that can reduce the transmission cost into the forwarding topology. Therefore, our algorithm has a lower transmission cost than existing NC-based protocols [4] [14] [19]. When implementing a routing protocol, nonetheless, we still need to face the choice between NC-based routing and single path routing. In this section, we study a few properties of our solution, which demonstrates the advantage of our NC-based transmission algorithm over traditional single path routing in terms of energy efficiency, i.e., transmission cost.

In traditional single path routing, it is the common sense that we always want to select the shortest path in the network. The term ”shortest” depends on different metrics or constraints we use, e.g., transmission cost, hop count, capacity and latency. However, when we use intra-flow network coding to tackle the forwarder selection problem in opportunistic routing to minimize the transmission cost, the first property we find for our solution is that the shortest (i.e., lowest cost) single path is not necessarily chosen into the transmission topology. This property is formally presented in the following theorem:

Theorem 2 *Given a node S with a candidate set FCS_S of M forwarders, the optimal forwarder set FS_S computed in Algorithm 2 does not always contain node A^* where $A^* \in FCS_S$ and $\frac{1}{P_{SA^*}} + C_{A^*} \leq \frac{1}{P_{SA_i}} + C_{A_i}$ for any $i \in FCS_S/\{A^*\}$.*

Proof The proof of this theorem is not complex. As long as we give an instance of node S with M forwarders that has the minimal cost transmission topology not including the lowest cost single path, we have the proof we need. Thus we build an instance in Figure 3.

In this instance, the lowest cost single path is $S \to A_3 \to T$ with a cost $\frac{1}{0.9} + \frac{1}{0.1} = 11.11$. After we run Algorithm 2, however, the optimal forwarder set we have is $FS_S = \{A_1, A_2\}$ because we have the following results:

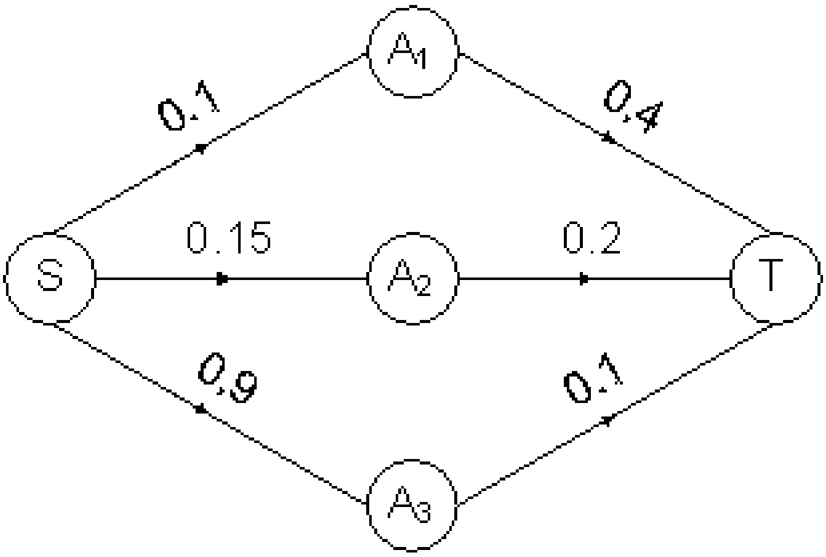

Figure 3: Routing braid v.s. single path routing

$$\begin{aligned} C_{\{A_1,A_2\}} &= \tfrac{1}{1-(1-0.1)(1-0.15)} \cdot [1+\tfrac{0.1}{0.4}+\tfrac{0.15(1-0.1)}{0.2}] \\ &= \tfrac{1}{0.235} \cdot (1+\tfrac{1}{4}+\tfrac{0.135}{0.2}) \\ &= 8.1915 \\ C_{\{A_1,A_2,A_3\}} &= \tfrac{1}{1-(1-0.1)(1-0.15)(1-0.9)} \cdot [1+\tfrac{0.1}{0.4}+\tfrac{0.15(1-0.1)}{0.2}+\tfrac{0.9(1-0.1)(1-0.15)}{0.1}] \\ &= \tfrac{1}{0.9235} \cdot (1+\tfrac{1}{4}+\tfrac{0.135}{0.2}+\tfrac{0.6885}{0.1}) \\ &= 9.5398 > C_{\{A_1,A_2\}} \end{aligned} \tag{13}$$

Using this instance, we finish our proof for this theorem.

From this example, it is also easy to see that the optimal transmission cost of NC-based transmission is lower than that of shortest single path routing. This further raises the question: will the minimal cost of NC-based transmission always be better than that of single path routing? To answer this question, we propose the following theorem:

Theorem 3 *Given a node S with a candidate set FCS_S of M forwarders, the optimal transmission cost C_S^* computed in Algorithm 2 is always lower than or equal to $\frac{1}{P_{SA^*}+C_{A^*}}$ where $A^* \in FCS_S$ and $\frac{1}{P_{SA^*}} + C_{A^*} \leq \frac{1}{P_{SA_i}} + C_{A_i}$ for any $i \in FCS_S/\{A^*\}$.*

Proof Through Theorem 2 we showed that the forwarder on the lowest cost single path is not always in the forwarder set computed in Algorithm 2. Therefore, we prove the correctness of this theorem 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 construction order of FS_S, we have the following inequity:

$$C_{A^*} \geq C_{A_j} \text{ for any } A_j \in FS_S \tag{14}$$

The only reason the algorithm does not add A^* into FS_S is because this operation will increase the total NC-based transmission cost. We denote $FS_S = \{A_1, A_2, \ldots, A_k\}$. This argument can be mathematically expressed as:

$$\begin{aligned}
C_S^* - C_{FS_S \cup \{A^*\}} &= \frac{1+\sum_{i=1}^{k}[C_{A_i}P_{SA_i}\prod_{j=1}^{i-1}(1-P_{SA_j})]}{1-\prod_{i=1}^{k}(1-P_{SA_i})} \\
&\quad - \frac{1+\sum_{i=1}^{k}[C_{A_i}P_{SA_i}\prod_{j=1}^{i-1}(1-P_{SA_j})]+C_{A^*}P_{SA^*}\prod_{i=1}^{k}(1-P_{SA_i^*})}{1-(1-P_{SA^*})\prod_{i=1}^{k}(1-P_{SA_i})} \\
&= \{1+\sum_{i=1}^{k}[C_{A_i}P_{SA_i}\prod_{j=1}^{i-1}(1-P_{SA_j})]\} \\
&\quad \cdot\{\frac{1}{1-\prod_{i=1}^{k}(1-P_{SA_i})} - \frac{1}{1-(1-P_{SA^*})\prod_{i=1}^{k}(1-P_{SA_i})}\} \\
&\quad - \frac{C_{A^*}P_{SA^*}\prod_{i=1}^{k}(1-P_{SA_i^*})}{1-(1-P_{SA^*})\prod_{i=1}^{k}(1-P_{SA_i})} \\
&= \frac{\{1+\sum_{i=1}^{k}[C_{A_i}P_{SA_i}\prod_{j=1}^{i-1}(1-P_{SA_j})]\}P_{SA^*}\prod_{i=1}^{k}(1-P_{SA_i^*})}{\{1-\prod_{i=1}^{k}(1-P_{SA_i})\}\{1-(1-P_{SA^*})\prod_{i=1}^{k}(1-P_{SA_i})\}} \\
&\quad - C_{A^*}\cdot\frac{P_{SA^*}\prod_{i=1}^{k}(1-P_{SA_i^*})}{1-(1-P_{SA^*})\prod_{i=1}^{k}(1-P_{SA_i})} \\
&= \frac{P_{SA^*}\prod_{i=1}^{k}(1-P_{SA_i^*})}{1-(1-P_{SA^*})\prod_{i=1}^{k}(1-P_{SA_i})}\cdot\{\frac{1+\sum_{i=1}^{k}[C_{A_i}P_{SA_i}\prod_{j=1}^{i-1}(1-P_{SA_j})]}{1-\prod_{i=1}^{k}(1-P_{SA_i})} - C_{A^*}\} \\
&< 0
\end{aligned} \tag{15}$$

From this inequity, we then conduct the following transformation:

$$\begin{aligned}
&\frac{1+\sum_{i=1}^{k}[C_{A_i}P_{SA_i}\prod_{j=1}^{i-1}(1-P_{SA_j})]}{1-\prod_{i=1}^{k}(1-P_{SA_i})} - C_{A^*}\} < 0 \\
\Leftrightarrow\quad & C_S^* - C_{A^*} < 0 \\
\Rightarrow\quad & C_S^* < \frac{1}{P_{SA^*}} + C_{A^*}
\end{aligned} \tag{16}$$

Therefore, when A^* is not selected into FS_S, the optimal NC-based transmission cost C_S^* is lower than the transmission cost of shortest single path routing.

2) $A^* \in FS_S$ In this case, we consider three scenarios:

a) If $FS_S = \{A^*\}$, it is clear that $C_S^* = \frac{1}{P_{SA^*}} + C_{A^*}$.

b) If $FS_S \neq \{A^*\}$ and A^* is the first node selected into FS_S, $C^* < \frac{1}{P_{SA^*}} + C_{A^*}$ is implied in the greedy forwarder selection process of Algorithm 2.

c) If $FS_S \neq \{A^*\}$ and A^* is not the first node selected into FS_S, it is straightforward that

$$\frac{1}{1-(1-P_{SA^*})\prod_i^{i\in FS_S}(1-P_{SA_i})} < \frac{1}{P_{SA^*}} \tag{17}$$

And it is implied in the greedy forwarder selection process that before adding A^* into FS_S, C_{A^*} is greater than or equal to the forwarding cost from the old FS_S to the destination. Therefore we still have $C^* < \frac{1}{P_{SA^*}} + C_{A^*}$ under this scenario.

Combining all different scenarios, we can reach the conclusion that the minimal cost of NC-based transmission is always smaller than or equal to the shortest single path routing. This completes our proof.

4 Protocol design and implementation

After we proposed a minimal cost NC-based routing algorithm and proved its advantage over traditional shortest single path routing through theoretical analysis, we move on to deploy this algorithm into resource-constrained wireless platforms, e.g. wireless sensor networks. Not only do we need to implement this core algorithm, we also need other components to build the whole routing protocol. When designing a NC-based routing protocol, there are three key challenges, which are:

1. For each node, which neighbor of it should be selected into the forwarder set?

2. For each node, how many times of broadcast it should conduct for a batch before it stops?

3. For each node, how fast it should broadcast a re-encoded packet for a batch?

To address these challenges, we propose the energy-efficient NC-based routing (EENCR) protocol to perform minimal cost NC-based transmission in wireless sensor networks. EENCR is a fully distributed routing protocol that runs on every node in the network. In this section, we present three key components of EENCR, each of which addresses one of the challenges listed above.

4.1 Routing engine

Run on each node, the routing engine component computes the optimal forwarder set for the current node, which address the first challenge. We design the routing engine in EENCR based on the 4-bit link estimator component and routing engine component of the collection tree protocol(CTP). Our routing engine is responsible for the following assignments:

(a) Estimate the single link reliability from the current node to each of its 1-hop neighbor;

(b) Compute and update the minimal cost of NC-based transmission from the current node to the designated destination based on the received transmission cost information from its neighbors;

(c) Broadcast the computed minimal cost and the forwarder set effective load table to all its 1-hop neighbor;

(d) Provide the optimal effective load information to the ACK component and the rate control component.

The key difference between the routing engine in EENCR and CTP is that instead of selecting only the neighbor on the shortest single path as the next hop forwarder, EENCR selects a set of neighbors into the forwarder set using Algorithm 2 such that the total transmission cost can be further reduced. In this way, we make use of the routing diversity of wireless communication to the max extent.

4.2 Modified NSB coded feedback

The routing engine component decides the forwarder set for the current node. In NC-based routing, each node needs to know when it can stop broadcasting to its forwarders. The condition for a node i to stop broadcast is that nodes in the forwarder set of i have collectively receive L_i linear independent packets, where L_i is the effective load information computed from the routing engine component.

The usual way node i gets information to decide when to stop transmitting is via the ACK feedback from nodes in FS_i. One naive approach is to make nodes in FS_i transmit ACK on a per-packet basis. However, this per-packet ACK cannot be used in EENCR due to two reasons.

- The total size of per-packet ACK for the whole effective load is too large. In practical network coding protocols with symbol size $GF(2^8)$ and batch size 8, each coding vector contains 8 bytes. If a forwarder j wants to convey the whole coding vector space it received from i, it will need K_i^j 8-byte vectors, which is too large for energy-constrained sensor networks.

- Sending back per-packet ACKs will introduce high-contention and communication overhead in the network, which reduces the energy-efficiency of the whole protocol.

One approach to avoid this high overhead is to use coded feedback. First proposed in [21], the null-space-based (NSB) coded feedback scheme is originally 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) was proposed in [14]. 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 designed to deploy in wireless mesh networks, where each node has a stronger computation power and larger storage space. It is hard to transplant it into sensor networks because:

- Compared to NSB, CCACK needs a much larger storage space to store M multiple hash metrics, where $M \geq 1$;

- To decrease the probability of false-positive, CCACK needs to run test algorithms M times, each of which with a different hash metrics;

Although CCACK can reduce the false-positive probability from $\frac{1}{2^8}$ to $(\frac{1}{2^8})^M$, it introduces both higher memory overhead and computation overhead. And when $M = 1$, the false-positive probability of CCACK is the same as NSB while having a more complex computation overhead. In fact, to overcome the collective-space problem in NC-based transmission, we only need a modified NSB ACK scheme (M-NSB) instead of the more complex CCACK.

We first elaborate how the original NSB ACK works. We denote the set of coding vectors received by node i to be B_i^r. When node i wants to broadcast about the feedback information of linear independent packets it currently has, it generates the feedback information as a vector z_i that satisfies:

$$z_i \cdot v = 0, \qquad \forall v \in B_i^r \tag{18}$$

Let V_i^r denote the linear space spanned by vectors in B_i^r. It is shown in [21] that:

Lemma 1 *With the above random construction of z_i, any vector $v' \in V_i^r$ must satisfy $z_i \cdot v' = 0$. And for any vector $v'' \notin V_i^r$, the probability of $z_i \cdot v'' = 0$ is $\frac{1}{2^8}$ when $GF(2^8)$ is used.*

The reason why NSB coded feedback may cause the collective-space problem is because NSB is not designed to convey the collective space of all downstream nodes but only the space relationship between the individual node pairs. To overcome this shortcoming while keeping the implementation at a low complexity level, we design the M-NSB coded feedback scheme. M-NSB has two different features from the original NSB:

1. Instead of generating z_i for set B_i^r, M-NSB generates z_i for set B_i^w, which is the coding vector set of all the re-encoded packets node i broadcasts. Then the condition z_i needs to satisfy becomes:

$$z_i \cdot v = 0, \qquad \forall v \in B_i^w \tag{19}$$

2. Node i stops broadcasting when there are L_i vectors in B_i^w are marked to be received by nodes in FS_i.

After a M-NSB ACK is generated, it is broadcast by the receiving node. M-NSB is different from CCACK in that M-NSB 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 EENCR, each node has its own effective load and packets received by the same node but from different senders will be viewed as 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. Therefore, M-NSB addresses the second challenge in designing NC-based routing protocols.

4.3 Rate control

In EENCR, the routing engine component provides the effective load information, and the M-NSB component provides the receipt status of re-encoded packets to the forwarder set. We then design a rate control component to help each node decide when to start the broadcast and how fast it should broadcast.

We first give the following definition of traffic flow:

Definition 2 *A traffic flow f is defined as a 5-tuple $f = (S, T, x, j, i)$ to represent a load of packets originated at node S and destined at T with batch index x, which is forwarded from sender j to forwarder i.*

At each non-destination node i, EENCR maintains an array $B_i^v(f)$ to store linear independent packets received for each flow. We also define a binary active-flow indicator I_f for each flow (S, x, j, i). I_f is set to be false by default and is updated to true only when one of the following two conditions is satisfied:

1. Node i is the first member of FS_j for flow f;

2. Node i receives more than $K_i(f) - L_i(f)$ linear independent packets from node j for flow f, where $K_i(f)$ is the number of linear independent packets i expected to receive from j for flow and $L_i(f)$ is the effective load assignment of node i for flow f.

Every time there is a transmission opportunity for node i, one active-flow is chosen in a round-robin fashion. A re-encoded packet is generated by selecting non-zero elements in $GF(2^8)$ as re-encoding vectors for packets in $B_i^v(f)$. Node i then broadcasts this re-encoded packet and adding the re-encoding vectors into $B_i^w(f)$. Once the forwarder set of i has received $L_i(f)$ linear independent packets, I_f is set to false and the array for flow f will be flushed.

5 Performance evaluation

To characterize the feasibility and effectiveness of network coding in improving the energy efficiency , we experimentally evaluate the performance of EENCR in this section. We first present the experimentation methodology and then the measurement results.

5.1 Methodology

Testbed. We use the *NetEye* wireless sensor network testbed at Wayne State University [1]. In this testbed, 130 TelosB motes are deployed, where every two closest neighboring motes are separated by 2 feet in an indoor environment. 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. Each of these TelosB motes is equipped with a 3dB signal attenuator and a 2.45GHz monopole antenna.

In our measurement study, we set the radio transmission power to be -7dBm (i.e., power level 15 in TinyOS) such that multihop networks can be created. And we use the default MAC protocol provided in TinyOS 2.x.

Protocols studied. To understand the impact of network coding in improving the energy efficiency of wireless sensor networks, we comparatively study the following protocols:

- *EENCR*: the distributed NC-based routing protocol we proposed, which selects the optimal forwarder set for each node to minimize the transmission cost;

- *CTP*: a state-of-the-art collection tree protocol designed for data collection in sensor networks [9];

- *MORE*: the first NC-based opportunistic routing protocol that fully explores the routing diversity in the network by letting each forwarder forward randomly coded packets;

- *CodeOR*: a NC-based opportunistic routing protocol that increases the concurrency of data flow by adding hop-by-hop ACK to MORE.

We implement all four protocols in TinyOS 2.x. Due to the constraints of memory space of TelosB motes, which is only 10 kilobytes, and the short data payload length in sensor network applications, we choose a batch size of 8 for network coding operation instead of the mostly used batch size of 32 in wireless mesh networks.

Performance metrics. For each protocol we study, we evaluate their 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*: number of forwarders selected to transmit a packet.

Different from the throughput metric used to evaluate the performance of NC-based routing protocols in [4] [19], in this study we use goodput instead. An information element is defined as **valid** if and only if it is linear independent to all elements that are in the same batch and received by the sink. And we do not study the routing diversity of CTP because its number of forwarders to transmit a packet is always one.

Traffic pattern. To experiment with both light and heavy traffic scenarios, we use two periodic data collection traffic patterns as follows:

- $S10$: out of all the 40 nodes in the networks, 10 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 EENCR, MORE and CodeOR, every consecutive 8 information elements compose a batch; this is to represent light traffic load scenarios.

- $S20$: same as $S10$ except that 20 nodes are selected as source nodes; this is to represent heavy traffic load scenarios.

5.2 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 the corresponding metrics.

5.2.1 Light data traffic

For the light traffic pattern $S10$, Figures 4 - 6 show the delivery reliability, delivery cost and goodput of different protocols. We found that EENCR and CTP provide high data delivery reliability (i.e., close to 100%) while MORE and CodeOR can only delivery 78% and 85% of the data to the sink on average. In the meantime, EENCR has a much lower delivery cost than CTP, i.e. a 26% reduction, in terms of average number of transmissions to deliver a packet but the delivery costs of MORE and CodeOR are around 400% and 300% of CTP respectively. Furthermore, EENCR enables a higher data goodput very close to the theoretical maximal value than all other three protocols.

The reasons for the inferior performance of MORE and CodeOR in our study are as follows:

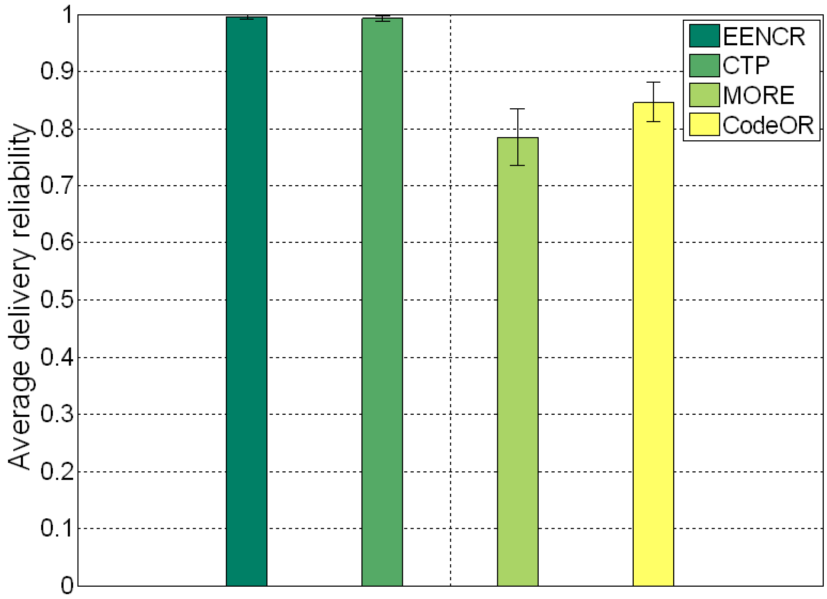

Figure 4: Delivery reliability: 10 sources

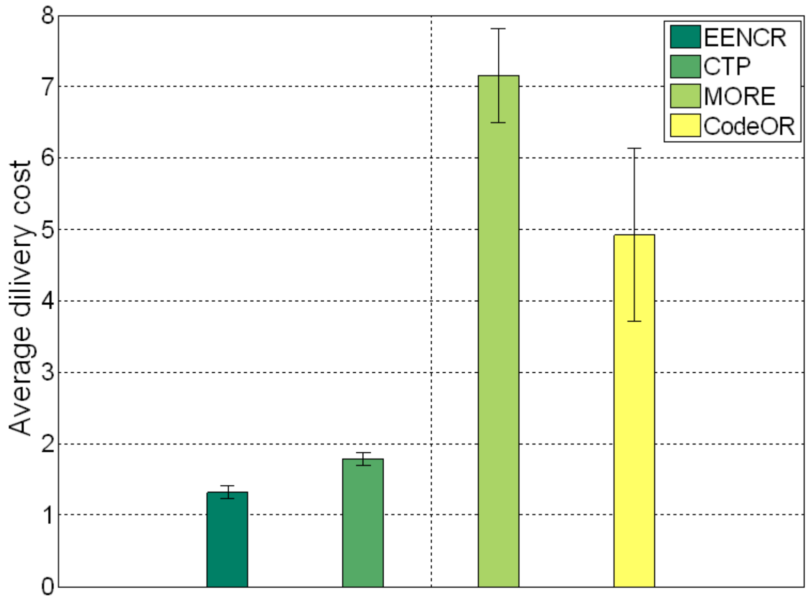

Figure 5: Delivery cost: 10 sources

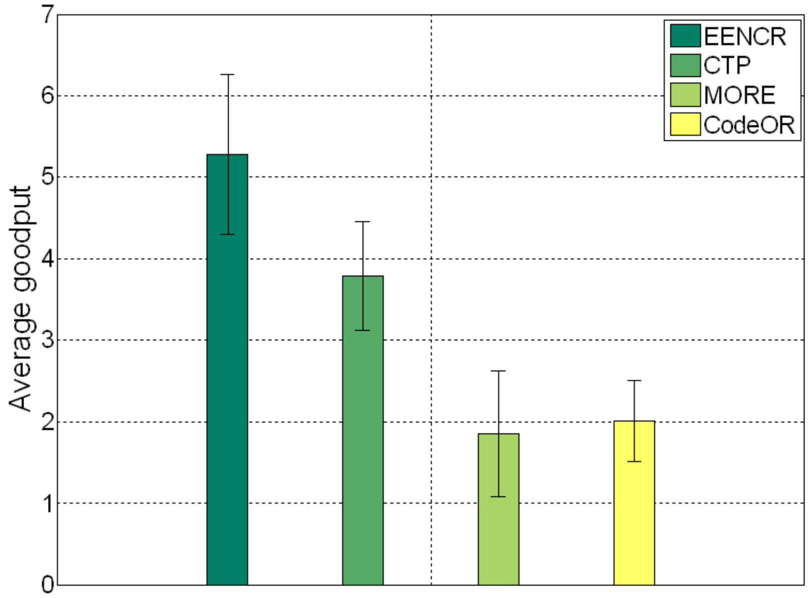

Figure 6: Goodput: 10 sources

1. The main design principle of MORE and CodeOR is to have all the forwarders encode and broadcast the packets they received. Although this principle makes full use of the spatial routing diversity for wireless networks, having all nodes in a network would significantly increase the contention of the network and compromising its performance. On the other hand, EENCR adopts an optimal greedy approach that only allows forwarders that can contribute in reducing the total transmission cost to get involved in the forwarding process. This strategy also helps reduce the contention in the network, which further improves EENCR's performance.

2. Both MORE and CodeOR rely heavily on the assumption of a reliable end-to-end ACK scheme to make source nodes and intermediated nodes stop broadcasting after the destination received enough coded packets for a certain batch. However, end-to-end ACKs tend to be unreliable, and it takes non-negligible time for all the nodes in the network to get an end-to-end ACK for a certain batch from the destination.

To elaborate on the above observations, we compare the number of forwarders selected in EENCR, MORE and CodeOR and summarize the results in Figure 9. It is shown in this figure

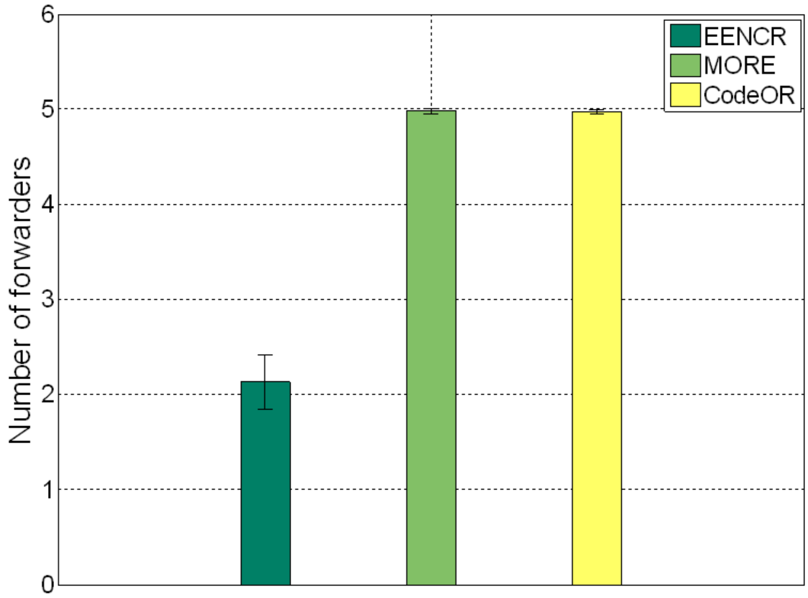

Figure 7: Routing diversity: 10 sources

that the average forwarders selected for each non-sink node in EENCR is around 2, but this number becomes 5 in MORE and CodeOR.

5.2.2 Heavy data traffic

To study the performance of EENCR in a more saturated network, we increase the number of sources to 20 to create a heavy traffic scenario $S20$. Figures 8 - 10 show the delivery reliability, delivery cost and goodput of different protocols. With heavier traffic in the network, EENCR is still able to provide a 98% data delivery reliability. Additionally, the reduction of EENCR compared to CTP has increased to 28%. This observation again is consistent with the design philosophy of EENCR. With heavier data traffic load in the network, the transmission cost of single path routing degrades. On the contrary, the transmission cost of EENCR still stays at a low level in that it fully explores and optimally leverages the wireless routing diversity in the network.

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

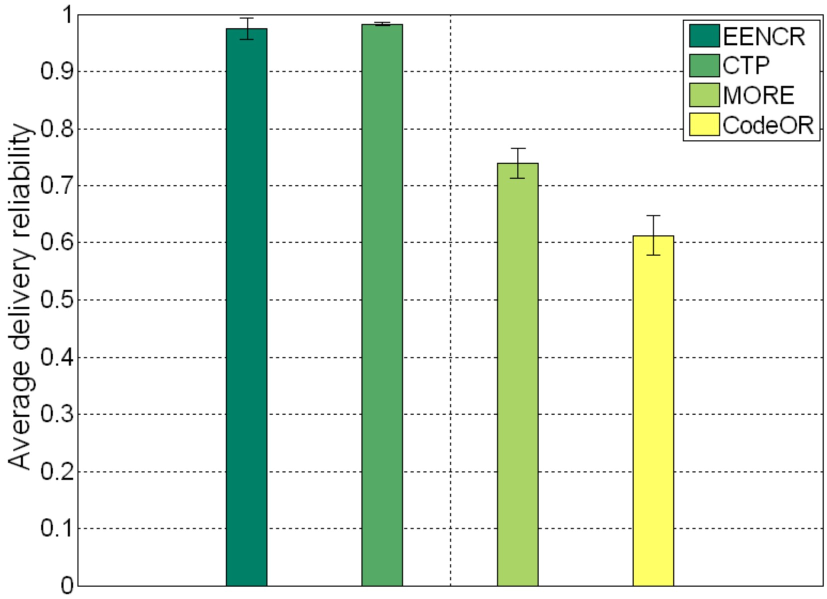

Figure 8: Delivery reliability: 20 sources

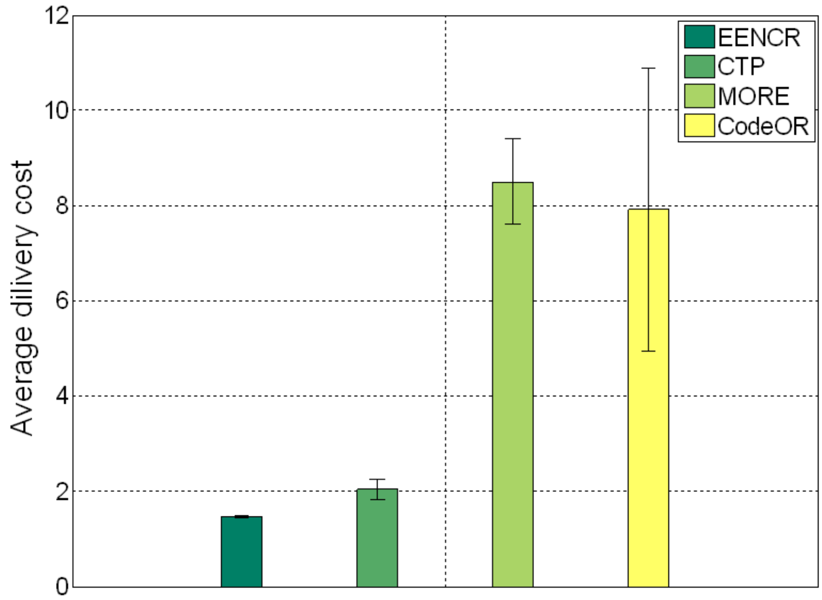

Figure 9: Delivery cost: 20 sources

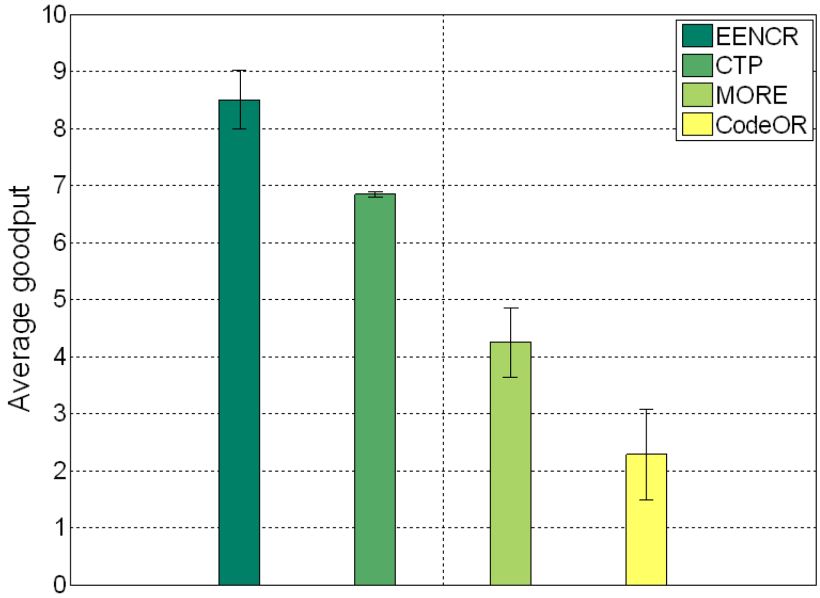

Figure 10: Goodput: 20 sources

CodeOR is even lower than MORE under $S20$. This is because CodeOR tries to increase the concurrency of the network by allowing multiple flows for the same source to be injected in the network. However, it still has all the forwarders in the network to encode and forward packets towards the destination, which would result in high contention and poor delivery performance in the network. Injecting too many flows in the network without considering the negative effects brought by allowing every forwarder to perform forwarding operation can be disastrous in a network with heavy traffic, as shown in our experiment results. We show the routing diversity in terms of average number of forwarders selected in these NC-based protocols in Figure 11. This observation demonstrates, from another perspective, that it is of great importance and necessity to choose forwarder sets in NC-based routing protocols carefully.

6 Related work

Network coding was first proposed for wired networks in the pioneering paper [2]. By mixing packets at intermediate nodes during the transmission, the bandwidth can be saved and

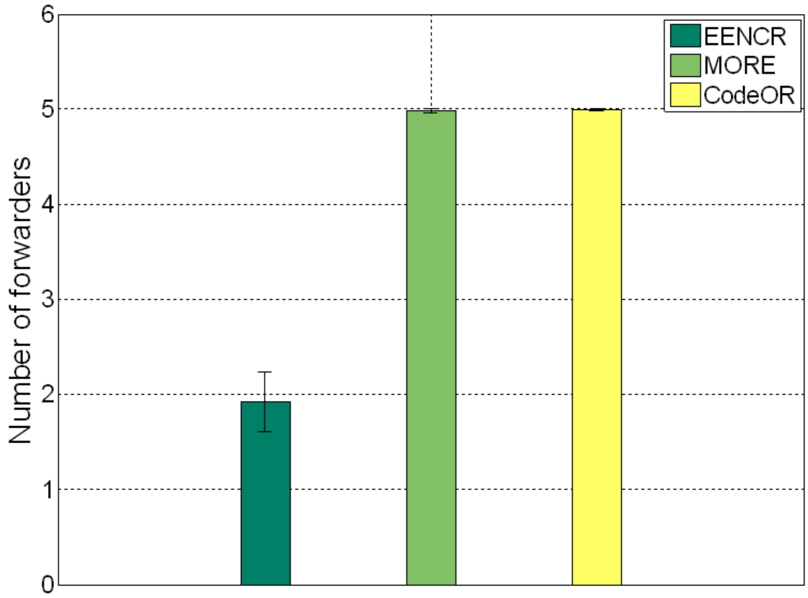

Figure 11: Routing diversity: 20 sources

therefore the throughput of the whole network can be significantly improved. During the past years, network coding has been one of the most popular research topics in computer networks. Different coding schemes are designed, categorized into linear network coding and non-linear network coding. Compared with linear network coding, non-linear network coding has been reported to outperform linear coding in several studies [17] [5] [16] [6]. Especially in [6], it is shown that there are multi-source network coding problems for which non-linear coding has a general better performance on throughput. Nevertheless, according to the analysis from [18], linear network coding can provide a performance close to the best possible throughput while only require a relative low complexity compared with the high complexity of non-linear coding.

Due to the broadcast nature in wireless communication, each intermediate node can receive redundant packets during the transmission in wireless networks. Network coding is one of the best choices to make use of these redundancies. By mixing redundant packets together and forwarding the mixed packet, the throughput of the wireless networks can be further improved. It is shown that linear coding functions can be designed randomly and independently at each node [11] [12]. Authors in these papers proposed a coding technique called random linear coding (RLC). Since RLC can be easily implemented in a distributed manner and it has a low complexity, it is widely used in wireless networks, including wireless sensor networks [10].

After network coding has been proved to be able to effectively use the overhearing redundancy in wireless environment, research on network coding in wireless networks has been following two different broad directions.

6.1 Network-coding-based multicast

Multicast has been well studied in wireless networks in the past few decades. Introducing network coding into multicast protocol, researchers find that the randomness of coded packets can effectively reduce the latency of multicast, therefore increase the network throughput.

Eryilmaz *et al.* [7] is the first work studying the delay performance gains from network coding. The authors study the problem on a wireless network model with one source and multiple receivers. Files are transferred from the source to receivers using network coding. The delay performance in this paper is defined as the average complete time of a file transmission. The authors study two different cases: 1) a file is broadcast to all receivers (broadcast case); 2) each receiver demands a different file (multiple unicast case). According to the theoretical analysis in this paper,there is a significant delay performance gain in both broadcast case and multiple unicast case via network coding, i.e., the average completion time is reduced.

Although network coding is proved to be able to provide average latency guarantee in [7], there is still a trade-off between the throughput and end-to-end latency for network coding in different wireless networks. Katabi *et al.* [8] used a simple example as follows to demonstrate this trade-off.

Suppose there are k packets needed to be sent from node A to B, link AB has a reliability of 50%. If node A sends these packets separately, it would require an expected number of transmission $4k$ including sending back k ACK packets. If all these packets are generated by A at the same time and therefore could be coded into k coded packets. Successfully sending these k coded packets would require an ETX of only $2k + 1$ including sending back only 1 ACK packet. If $k/2$ packets are generated first and has to be sent to B before the other $k/2$ packets are generated, these k packets could only be coded into two groups with $k/2$ coded packets each. The whole ETX for this transmission scheme is $2k + 2$ including sending back 2 ACK packets.

Zhang *et al.* [23] investigate the benefits of using Random Linear Coding (RLC) for unicast communications in a mobile Disruption Tolerant Network (DTN) under epidemic routing. In this paper, the authors propose the following coding and transmitting scheme: DTN nodes store and then forward random linear combinations of packets as they encounter other DTN nodes. The simulation results show that when there is one single file composed of several packets propagating in the network, when bandwidth is constrained, applying intra-flow RLC over packets can improve the delivery delay to deliver the whole file, and there is more improvement when the buffer in each node is limited. When there are multiple files propagating in the network, simulations results show that intra-flow RLC offers only slight improvement over the non-coded scheme when only bandwidth is constrained, but more significant improvement when both bandwidth and buffers are constrained.

The work in the above paragraph studies the benefits of network coding in DTN by a simulation based approach. Different from [23], Lin *et al.* [20] study this problem in a theoretical analysis framework. The theoretical analysis achieves similar conclusions as those in [23]. Based on the analysis, the authors also design a priority coding protocol, in which packets in the same file are divided into different groups with priorities and packets with higher priority would be coded and transmitted first. When the destination receives all coded packets for a certain level, it notifies the whole network and the source so that the same packets stored in the network will be dropped to further increase the performance of the network.

In both [23] and [20], the authors do not consider interferences in the network, which is reasonable only for sparse networks. Zhang *et al.* [22] conduct an analysis on the throughput-delay tradeoffs in mobile ad hoc networks (MANETs) with network coding, and compare results in the situation where only replication and forwarding are allowed in each node. The network model is built on both fast mobility model (i.i.d. mobility model) and slow mobility model (random walk model). The authors propose a k-hop relay scheme in a n -node MANET using RLC in MANETs and prove the trade-off between throughput and delay of the proposed scheme under two mobility models. Under fast mobility model, where $k = \Theta(\log n)$, the throughput $T(n) = \Theta(1/n)$ and the average delay $D(n) = \Theta(\log n)$, where $T(n)$ represents throughput and $D(n)$ represents average delay. Under the slow mobility mode, where $k = \Theta(\sqrt{n})$, $T(n) = \Theta(1/n)$ and $D(n) = \Theta(\sqrt{n})$. This is the first work to study the trade-off between throughput and delay using RLC in MANETs. However, this study still uses the average delay as the metric instead of putting hard latency constraints on the analysis.

Katti *et al.* [13] propose COPE, a new architecture for wireless mesh networks. It is the first network coding that is implemented with the current network stack seamlessly. In the design of COPE, only inter-flow network coding is concerned. That means packets headed to the same next hop or generated by the same source cannot be encoded together under COPE. And COPE adopts an opportunistic coding scheme, which does not delay packets' transmissions for further coding opportunity. According to the theoretical analysis, not only can network coding bring a significant improvement on throughput, but also the MAC layer protocol can also improve the network throughput when it is combined with coding technique. COPE is implemented on a 20-node wireless network testbed. The experiment results show that COPE can increase the throughput of wireless mesh networks without modifying routing or higher layers.

6.2 Network-coding-based opportunistic routing

Other than network coding, opportunistic routing is another technique that fully explores the diversity of the broadcast nature in wireless communication. ExOR is the first opportunistic routing protocol and was proposed in [3]. Since then, extensive work has been conducted to further improve the forwarder candidate selection process in opportunistic routing. However, the essential component in opportunistic routing protocols incurs heavy communication cost of node coordination and requires a delicately designed MAC protocol.

As a continuous research of [3][13], Chachulski *et al.* [4] integrated intra-flow RLC and the opportunistic routing protocol in [3] to develop a new routing protocol called MORE in wireless mesh networks. The contribution of MORE is multi-dimensional. First, it makes use of the broadcast property of wireless communication to improve the network throughput without modifying the existing MAC layer, e.g., 802.11. Secondly, it adopts RLC for intra-flow network coding. RLC has a low complexity and is is easy to implement in a distributed system. Therefore, the network throughput is further improved. Thirdly, both the memory overhead and the header overhead are bounded within a reasonable range. MORE is also evaluated in a 20-node testbed and it outperform ExOR in both unicast and multicast traffic flow with a higher throughput.

Quite a few new protocols has been built based on MORE to further improve the throughput of NC-based opportunistic routing [19] [14] [10] [15] [24]. The basic idea of these studies is the natural combination of opportunistic routing and network coding because they both made use of the broadcast nature of wireless transmission. Koutsonikolas *et al.* [15] propose another intra-flow network coding architecture called Pacifier. Pacifier builds an efficient multicast tree and extends it to opportunistic overhearing. Then it applies intra-flow RLC technique to ensure the reliability. Both these two steps are similar with MORE. Besides these two components, Pacifier also applies a source rate control module to avoid the congestion in the network. Most importantly, Pacifier solves the ”crying baby” problem by having the source send batches of packets in a round-robin fashion. Not only large scale simulations but also a series of experiments in a 22-node wireless testbed show that Pacifier have a large improvement on average throughput compared with MORE. Similar to Pacifier, [10] proposed Rateless Deluge, the first implementation of NC-based opportunistic routing protocol in wireless sensor networks.

Zhu *et al.* [24] propose a hybrid coding scheme that does inter-flow coding first and intra-flow coding later. In the proposed scheme, packets are first encoded following the same coding scheme adopted by COPE. Then the encoded packets are divided into different batches. Encoded packets in the same batch are further encoded following the same coding scheme adopted by MORE. During the transmission, the whole system uses a multiple-path transmitting scheme to further improve the network throughput. The authors do a theoretical analysis on their proposed coding scheme in a simple wireless network model. Compared with COPE, the hybrid coding scheme has a significant improvement on both throughput and reliability in this network model. However, simulation or experiments are needed to further testify the efficiency of this hybrid scheme.

To further improve the throughput of wireless networks, Lin *et al.* [19] make use of hop-by-hop ACK and sliding window to allow different segments of packets to be transmitted in the network concurrently (CodeOR). However, it still adopts offline ETX metric to decide how many coded packets to transmit to ensure the end-to-end decodability. To be adaptive to the dynamic of wireless links, Koutsonikolas *et al.* [14] uses a Cumulative Coded ACK(CCACK) scheme to allow nodes to notifying their upstream nodes that they have received enough coded packets in a simple and low overhead way. The throughput of CCACK is shown to be 45% better than MORE. [14] is the most closely work related to our problem. The cumulative coded ACK scheme gives a good solution to the problem ”when should a sender stop broadcasting”. However, CCACK’s major objective is to minimize the broadcast cost at each sender/forwarder. This approach cannot give a global minimization on transmission cost for NC-based opportunistic routing. Furthermore, CCACK requires a high memory space and a relatively complex computation process, which is not suitable for resource-constrained sensing networks.

7 Concluding remarks

NC-based routing has drawn the interests of many researchers in wireless community. In this section we studied the minimal cost NC-based routing problem. We proposed the first effective load based mathematical framework to compute the cost of NC-based routing for a given topology. To the best of our knowledge, this is the first successful attempt towards measuring the energy consumption of NC-based routing. Our solution provides a formal theoretical method to measure the transmission cost of intra flow network coding routing protocols.

Based on this framework, we then studies the open problem of computing the optimal transmission cost of NC-based transmission and the corresponding routing braid. We were able to derive a distributed polynomial-time greedy algorithm for this problem an proved its optimality. We further studied the property of this algorithm and showed that the optimal routing braid does not necessarily contains the shortest single path route as expected in traditional routing and opportunistic routing protocols. Plus, we proved that the upper bound of the energy consumption for optimal routing braid is the same as that of single path routing in terms of expected number of transmissions.

Furthermore, we proposed EENCR, an energy-efficient NC-based routing protocol for resource-constrained sensor networks. In EENCR, we adopted the 4-bit link estimator[9] and our minimal cost forwarder set selection algorithm in the routing engine component. We then developed M-NSB, a coded feedback scheme without near-zero additional communication overhead and designed a rate control component to avoid the energy waste caused by unnecessary broadcast. EENCR incorporated the design philosophy of CTP[9], a state-of-art single path routing protocol in sensor networks, so that the complexity of protocol is maintained at a low level, which is of great importance and favorable on low-power distributed platforms, e.g., TelosB sensors. Experiment results of EENCR on the NetEye testbed showed that EENCR yields a high reliability as CTP, and has a transmission cost that is only around 72-75% of CTP. In the meantime, the goodput of EENCR is significantly improved from MORE and CodeOR because it adaptively selects the forwarders instead of utilizing the whole forwarder candidate set.

References

[1] NetEye testbed. http://neteye.cs.wayne.edu/neteye/home.php.

[2] R. Ahlswede, Ning Cai, S.-Y.R. Li, and R.W. Yeung. Network information flow. *Information Theory, IEEE Transactions on*, 46(4):1204 –1216, July 2000.

[3] Sanjit Biswas and Robert Morris. ExOR: Opportunistic multi-hop routing for wireless networks. In *ACM SIGCOMM*, 2005.

[4] Szymon Chachulski, Michael Jennings, Sachin Katti, and Dina Katabi. Trading structure for randomness in wireless opportunistic routing. In *Proceedings of the 2007 conference on Applications, technologies, architectures, and protocols for computer communications*, SIGCOMM '07, pages 169–180, New York, NY, USA, 2007. ACM.

[5] R. Dougherty, C. Freiling, and K. Zeger. Linearity and solvability in multicast networks. *Information Theory, IEEE Transactions on*, 50(10):2243–2256, Oct. 2004.

[6] R. Dougherty, C. Freiling, and K. Zeger. Insufficiency of linear coding in network information flow. *Information Theory, IEEE Transactions on*, 51(8):2745–2759, Aug. 2005.

[7] A. Eryilmaz, A. Ozdaglar, and M. Medard. On delay performance gains from network coding. In *Information Sciences and Systems, 2006 40th Annual Conference on*, pages 864 –870, 2006.

[8] Christina Fragouli, Dina Katabi, Athina Markopoulou, Muriel Medard, and Hariharan Rahul. Wireless network coding: Opportunities & challenges. In *Military Communications Conference, 2007. MILCOM 2007. IEEE*, pages 1–8, Oct. 2007.

[9] Omprakash Gnawali, Rodrigo Fonseca, Kyle Jamieson, David Moss, and Philip Levis. Collection tree protocol. In *ACM SenSys*, 2009.

[10] Andrew Hagedorn, David Starobinski, and Ari Trachtenberg. Rateless deluge: Over-the-air programming of wireless sensor networks using random linear codes. In *Proceedings*

of the 7th international conference on Information processing in sensor networks, IPSN '08, pages 457–466, Washington, DC, USA, 2008. IEEE Computer Society.

[11] T. Ho, R. Koetter, M. Medard, D.R. Karger, and M. Effros. The benefits of coding over routing in a randomized setting. In *Information Theory, 2003. Proceedings. IEEE International Symposium on*, pages 442–, June-4 July 2003.

[12] S. Jaggi, P.A. Chou, and K. Jain. Low complexity algebraic multicast network codes. In *Information Theory, 2003. Proceedings. IEEE International Symposium on*, pages 368–368, June-4 July 2003.

[13] Sachin Katti, Hariharan Rahul, Wenjun Hu, Dina Katabi, Muriel Médard, and Jon Crowcroft. Xors in the air: practical wireless network coding. *IEEE/ACM Trans. Netw.*, 16(3):497–510, 2008.

[14] D. Koutsonikolas, Chih-Chun Wang, and Y.C. Hu. Ccack: Efficient network coding based opportunistic routing through cumulative coded acknowledgments. In *INFOCOM, 2010 Proceedings IEEE*, pages 1 –9, 2010.

[15] Dimitrios Koutsonikolas, Y. Charlie Hu, and Chih-Chun Wang. Pacifier: High-throughput, reliable multicast without.

[16] April Rasala Lehman. *Network coding*. PhD thesis, Cambridge, MA, USA, 2005. Supervisor-Sudan, Madhu.

[17] April Rasala Lehman and Eric Lehman. Complexity classification of network information flow problems. In *SODA '04: Proceedings of the fifteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 142–150, Philadelphia, PA, USA, 2004. Society for Industrial and Applied Mathematics.

[18] S.-Y.R. Li, R.W. Yeung, and Ning Cai. Linear network coding. *Information Theory, IEEE Transactions on*, 49(2):371–381, Feb. 2003.

[19] Yunfeng Lin, Baochun Li, and Ben Liang. Codeor: Opportunistic routing in wireless mesh networks with segmented network coding. In *Network Protocols, 2008. ICNP 2008. IEEE International Conference on*, pages 13 –22, 2008.

[20] Yunfeng Lin, Baochun Li, and Ben Liang. Stochastic analysis of network coding in epidemic routing. *Selected Areas in Communications, IEEE Journal on*, 26(5):794–808, June 2008.

[21] Joon-Sang Park, M. Gerla, D.S. Lun, Y. Yi, and M. Medard. Codecast: a network-coding-based ad hoc multicast protocol. *Wireless Communications, IEEE*, 13(5):76–81, 2006.

[22] Chi Zhang, Yuguang Fang, and Xiaoyan Zhu. Throughput-delay tradeoffs in large-scale manets with network coding. In *INFOCOM 2009, IEEE*, pages 199–207, April 2009.

[23] Xiaolan Zhang, G. Neglia, J. Kurose, and D. Towsley. On the benefits of random linear coding for unicast applications in disruption tolerant networks. In *Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks, 2006 4th International Symposium on*, pages 1–7, April 2006.

[24] Xiaoyan Zhu, Hao Yue, Fang Yuguang, and Yumin Wang. A batched network coding scheme for wireless networks. *Wireless Networks*, 15(8):1152–1164, Nov 2009.