Cluster-based Cooperative Communication with Network Coding in Wireless Networks

Zygmunt J. Haas
School of Electrical and Computer Engineering
Cornell University
Ithaca, NY 14850, U.S.A.
Email: haas@ece.cornell.edu

Tuan-Che Chen
School of Electrical and Computer Engineering
Cornell University
Ithaca, NY 14850, U.S.A.
Email: tc397@cornell.edu

***Abstract*—Cooperative communication is a promising way to reduce probability of packet loss. The massive deployment of nodes in wireless sensor network renders such networks especially attractive for exploiting the advantage of *cooperative diversity*. Similarly, when used appropriately, *network coding* could also improve the probability of correct reception. In this paper, we introduce the cluster-based Cooperative Coding (CC) protocol, which is based on the integration of cooperative communication and network coding. In particular, in the CC protocol, network nodes are grouped into multiple clusters and nodes within the same cluster cooperate in transmitting and receiving packets. Such an integration reduces the amount of redundant information being forwarded to ensure high probability of correct end-to-end reception, when link-level retransmission of erroneous packets is not allowed (i.e., no link-level feedback). In particular, our analysis shows how to optimize the performance of the network by properly sizing the clusters. Compared to schemes without cooperation (whether with or without network coding), our simulation results demonstrate the significant performance improvement of the proposed scheme.**

I. INTRODUCTION

In multi-hop wireless sensor networks, the information from the source to the destination is relayed by intermediate nodes. Traditionally, the routing protocols choose a path - a sequence of nodes between the source and the destination - and then forward packets along the path. To combat the link-level packet loss and to avoid significant end-to-end throughput degradation, networks use link-level retransmissions. However, due to correlation of errors in retransmitted packets especially in wireless networks, retransmission is often quite ineffective. It could also be quite inefficient, leading to significant waste of network capacity and energy, and considerably increasing the end-to-end delay. Thus, in numerous instances, such as real-time traffic for example, link-level retransmission may not be the right approach for increasing the end-to-end transmission reliability.

In contract, *cooperative communication* [1] has recently received significant attention as a way to improve the reliability of wireless links. For instance, the cooperative scheme in [2] suggests that the traditional routing may not be the best approach. Cooperative communication exploits the broadcast nature of wireless communications, where with a single transmission, a number of cooperating nodes receive and relay the data. Due to spatial *diversity* a receiver can then combine multiple relayed signals **(diversity combining)** or choose the best signal **(selection diversity)** at the physical layer to improve the overall channel quality [3]. Moreover, due to reduced time- and space-correlation of the transmission fading, the overall reliability of the received signal is increased. Depending on the type of diversity used at the physical layer, radios may need to be able to synchronize reception of multiple signals.

In cooperative communication, clustering could be used to group nodes which are located close to each other. The massive deployment of the nodes in wireless sensor network provides an effective scenario for node clustering. All nodes in a cluster cooperate to transmit and receive packets to/from other cooperative clusters. Compared with other schemes, the cluster-based approach reduces the complexity of resource management of the cooperation among the cluster's nodes.

Recent research in network coding has revealed its potential in increasing the capacity of wired and wireless networks. The capacity gain is achieved through coding of information received from multiple sources [4].

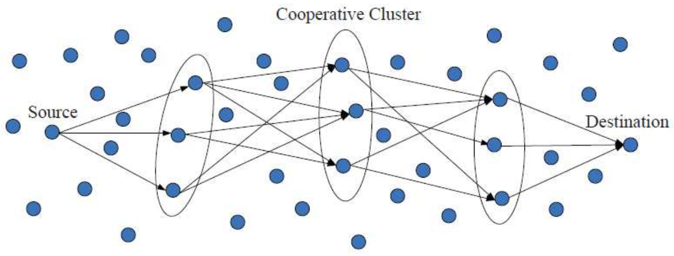


Figure 1: Example of cooperative clusters in a wireless network

In this paper, we integrate network coding to introduce the cluster-based Cooperative Coding (*CC*) protocol. Unlike cooperative communication schemes at the physical layer, CC integrates coding and cooperative diversity at the link layer, which could be implemented with standard radio hardware. Indeed, it is natural to explore the combining of network coding with cooperative diversity at the link layer, because in cooperative communication the broadcasting property of the wireless medium allows a node to overhear packets from multiple relays. The relay nodes can encode overheard information from different sources and forward the coded packet to other nodes. When enough information is received at the destination nodes, the destination nodes can then recover the original packets.

In our model of CC, there are multiple nodes in the receiver and in the sender clusters of each hop. Fig. 1 illustrates an example of cooperative transmissions from the source to the destination through multiple clusters, where packets are relayed from a cluster to a cluster. We assume that the intra-cluster distances are much smaller compared with the inter-cluster distances. As opposed to the traditional case in which each hop is composed of a point-to-point link, in cooperative transmission, each hop is replaced with many-to-many links. A node that hears packet transmission from the nodes in the previous cluster will relay the packet to the next cluster towards the destination. Therefore, the routing path can be represented as having a "width," which is determined by the number of nodes in a cluster. This cooperation in relaying the packets increases the probability that the packet will reach the destination. To achieve this goal, in this paper, a simple clustering and medium access control are introduced based on the work in [9]. In addition, to reduce the number of packet transmissions, CC randomly mixes the received packets and relays the coded packet by cooperating with the nodes within the same cluster. Thus a more reliable communication can be achieved. We show how to compute the number of cooperating nodes in a cluster as to optimize the end-to-end performance.

We also compare our proposed CC protocol with schemes that do not employ cooperation, whether with or without network coding. Our analytical results, validated by simulations, show that CC leads to significant performance improvement of throughput and of successful packet reception probability. Thus, we demonstrate that the CC protocol can exploit the integration of cluster cooperation and network coding in improving the network performance.

The paper is organized as follows. In Section II, we review the related work. Section III presents an example of how packet forwarding could benefit from cooperation and network coding. The proposed CC protocol is then introduced in Section IV. Section V presents a mathematical analysis to calculate the proper number of nodes in a cluster for optimizing the end-to-end performance. Section VII discusses performance evaluation, followed by conclusion in Section VIII.

II. RELATED WORK

A. *Cooperative Communication*

In the wireless network environment, where transmission is subject to loss, exploiting multi-user diversity is one of the techniques to combat the transmission impairments. At the physical layer, nodes overhearing a transmission simultaneously relay the signal. Such cooperative communication was investigated in numerous works (e.g., [3], [5], [6]), and either "amplifying-and-forwarding" or "decoding-encoding-and-forwarding" is performed by the relay nodes.

Several works have proposed cluster-based cooperative communication. In [7]–[9], the cluster design and corresponding energy conservation are investigated. Given energy constraint for each link, [10] proposes a scheme to minimize end-to-end outage probability. The CC scheme builds on the idea of these cluster cooperation schemes, but adopts a fundamentally different approach – it uses flow-based network coding on the link layer. Also, in contrast with most cooperative diversity approaches, the CC scheme does not require synchronized signal transmissions.

Opportunistic routing is a technique that realizes some gains of cooperative diversity at the link layer. The testbed of the ExOR protocol [11] was first shown to improve performance over the traditional deterministic forwarding. However, in ExOR, to prevent medium collisions, a strict transmission scheduling is imposed at the cost of reduced spatial reuse. The MORE protocol [12] addresses this issue and further improves the throughput by using network coding. In MORE, there is no particular next-hop for opportunistic routing. Once a node overhears a coded packet, it is involved in forwarding the packet. In contrast, CC builds a structural way to transmit coded packets by grouping the nodes into the cluster. The next-hop is limited to a finite group of nodes. Therefore, unnecessary transmissions are avoided, and the complexity for resource management of nodes is reduced. The CC scheme takes the advantages of cluster-based forwarding, spatial diversity, and network coding.

B. *Network Coding*

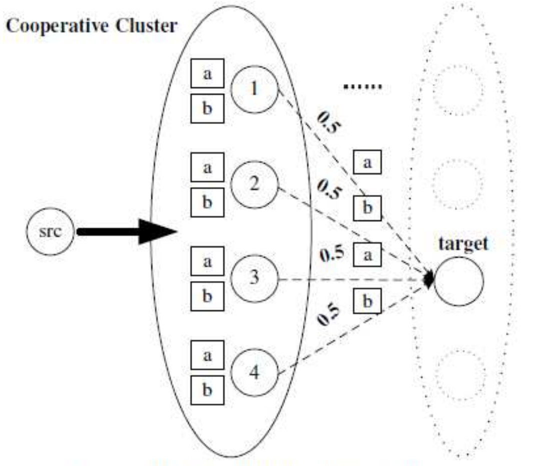
As opposed to traditional networks with single relay hop, in network coding intermediate nodes intelligently mix the received packets, so that the resulting transmitted packets contain information of multiple messages. One of the first works that studied network coding was the paper by Ahlswede et al. [4], which analyzed capacity bound of networks. Several papers (e.g., [4], [13], [14]) show that network coding can achieve the maximum multicast capacity in wired network. In the context of wireless networks, network coding was proposed to improve the performance of multicast [15] and broadcast [16]. Testbed experiments were conducted to demonstrate the throughput gain for unicast applications [17]. Through theoretical analysis, [18] quantified the potential throughput gain of coding-aware routing.

III. MOTIVATION

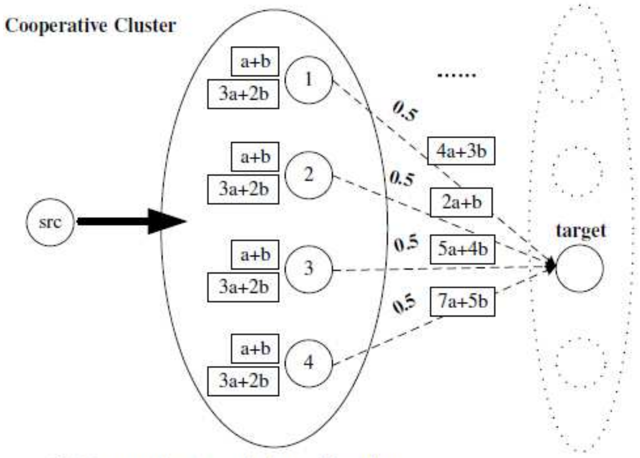
To justify the integration of cooperative communication with network coding, this section presents an example that demonstrates the improvement in performance of such an integrated scheme. Consider Fig. 2, where the source node *(src)* attempts to deliver two packets, *a* and *b*, to a destination. Assume that all the nodes in the cooperative cluster received two packets and that the probability of a successful transmission over any link to a target node in the next cluster is *0.5*. Note that the target node can be one of the nodes in the next cluster or the destination node. As discussed in Section I, we assume that link-layer retransmissions are not feasible.

In Fig. 2(a), packets *a* and *b* have been both received (overheard) by all the nodes in the cluster. To maximize the probability of reception of both packets, the nodes in the cluster transmit each packet the same number of times. Thus, for example, nodes 1, 2, 3, and 4 transmit packets *a*, *b*, *a*, and *b*, respectively. In this case, the probability that both packets *a* and *b* are received by the target is $(1-(1-0.5)^2) \times (1-(1-0.5)^2) = 0.5625$. With network coding, as in Fig. 2(b), each node in the cooperative cluster having overheard the coded packets, (*a*+*b*) and (*3a*+*2b*), from the previous cluster (the *src* node in this case), creates and transmits a linear combinations of the received coded packets. In this way, as long as the target receives two independent coded packets, the two original packets *a* and *b* can be recovered. Thus, the probability that both packets *a* and *b* can be acquired by the target is $(1-(1-0.5)^4-4\times(1-0.5)^4) = 0.6875$, which is higher than the corresponding probability of the non-coding scheme in Fig.

2(a). Network coding offers an elegant solution to improve reliability in cluster-based forwarding. Without network coding each relayed transmission contains only information of one original packet. With network coding loss of some of the transmissions could be compensated for, because each relayed transmission contains linear combination of some of the original packets. Hence, the original packets could be recovered from the correctly received transmissions. For example, as in Fig. 2(b), assume that node 1, node 2, node 3, and node 4 create and transmit the linear combinations (*4a+3b*), (*2a+b*), (*5a+4b*), and (*7a+5b*), respectively. Despite the fact that one or even two of the transmitted linear combinations are lost in transmission, the target node can still recover the original packets *a* and *b* from the correctly received linear combinations. On the other hand, in Fig. 2(a), when the transmission of nodes 1 and 3 (or transmissions of nodes 2 and 4) are lost, packet *a* (or *b*) cannot be recovered by the target node.



(a) Cooperative transmission without coding



(b) Cooperative transmission with coding

Figure 2: Example of the benefits of network coding and cooperation

Of course, the performance of a protocol that build upon the above cooperative and coded communicated depends on a number of parameters. One such key parameter is the size of the cooperative cluster. Consider again the example of Fig. 2(b), where the successful receipt probability is 0.6875. To achieve higher performance gains, more nodes are needed to help relay the packets. Furthermore, not all the combinations transmitted by nodes in a cluster need to be successfully retrieved by any single relay in the next cluster. The multiple relays with different coded packets in the same cluster can further cooperate to forward different coded packets as long as sufficient information is relayed to the next cluster. In Section V, we address the selection of the cluster size, so as to ensure high decoding probability at the destination.

IV. THE CLUSTER-BASED COOPERATIVE CODING PROTOCOL

In this section, we present the proposed CC protocol. We first introduce how CC incorporates routing and medium access control in a cooperative manner, and how to apply network coding as part of the packet forwarding operation. Then, we describe how the cooperative clusters are formed.

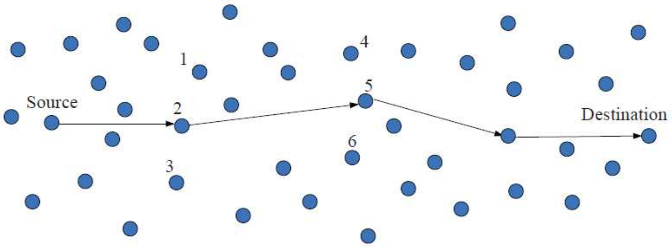


Figure 3: “One-node-width” path

A. Routing and Medium Access Control with Network Coding

In CC, a “one-node-width” path is first discovered between the source and the destination, as illustrated in Fig. 3. Such an initial path can be found by traditional routing protocols; e.g., AODV (Ad hoc On Demand Distance Vector) [19] or DSR (Dynamic Source Routing) [20]. The energy transmission for each link can be used as the link’s cost to discover minimal cost path. Next, the nodes on the initial path become cluster heads and recruit other nearby nodes to form clusters. CC is designed to transmit data in a block of coded packets from a cluster to a cluster. Recruiting of cluster nodes is done per block of forwarded packets. Recruiting of nodes of the next cluster commences once all the coded packets of the block are received by the previous cluster. Then, the newly generated coded packets of the block are forwarded, where the nodes within the sending and the receiving clusters cooperate in transmitting and receiving.

1) The Source: The source node partitions the data into blocks of *m* packets. The *m* uncoded packets in a block are called *native packets*. A native packet is denoted by x_i, ∈ {1, 2 . . . ,*m*}. The source transmits coded packets to the nodes in the next cluster. A coded packet x'_j is a linear combination of the native packets, generated as:

$$x'_j = \sum_{i=1}^{m} c_{ij} x_i$$

where the c_{ji}'s are the coefficients picked randomly, and the addition and multiplication are operations over a Galois Field, GF(2^q). We embed a code vector, $\vec{c}_j = (c_{j1}, c_{j2}, \ldots, c_{jm})$, and the block id, into the x'_j packet's header. The source maintains a counter with some initial value m', where $m' > m$. Each time the source transmits a coded packet, the counter is decreased by 1. The source keeps transmitting randomly coded packets until the counter reaches zero.

2) Recruiting and Forwarding: The recruiting and forwarding operations run per hop from the source to the destination. The cluster head of a receiving cluster initiates the construction of the next cluster after it has received the coded packets of the same block from the sending cluster. The receiving cluster now becomes the current sending cluster consisting of the same nodes which received the packets from the previous hop.

For each node of the sending cluster, the sending cluster head first schedules the specific time when it can transmit the coded packet to the receiving cluster. The receiving cluster head then recruits adjacent nodes to form the receiving cluster, and selects the nodes with higher cost C_j . We define C_j for node j, which may possibly be recruited for the receiving cluster, as: $C_j = \sum_{\forall\ node\ i \in\ sending\ cluster} I_{ij}(1-p_{ij})$, where p_{ij} is the loss probability of sending a packet from node i to node j, and I_{ij} is the indication function whether node j is available for receiving packets when node i in the sending cluster is transmitting. Node j may not be available because it has been scheduled to transmit or receive packets for other cluster of a different route. The p_{ij}-s are periodically evaluated by a node for all of its neighbors via ping probes. I_{ij} depends on how the sending cluster head schedules the nodes to transmit packets. Therefore, the nodes which are likely to receive more packets from the sending cluster are chosen to form the receiving cluster.

Assuming that a node has received the coded packets x's, the new coded packet can be generated as $x'' = \sum_{j=1}^{m} c_j x'_j$, where c_j'-s are random numbers chosen from GF(2^q). In this way, the x'' is also a random linear combination of the native packets, since

$$x'' = \sum_{j=1}^{m} c_j \left(\sum_{i=1}^{m} c_{ji} x_i\right) = \sum_{i=1}^{m} \left(\sum_{j=1}^{m} c_j c_{ji}\right) x_i = \sum_{i=1}^{m} g_i x_i .$$

Like the source node, when forwarding the coded packet, a node will embed the new code vector, $\vec{g}_j = (g_1, g_2, \ldots, g_m)$ within the x'' packet's header.

3) *At the destination:* When the destination obtains a coded packet, it will first check if the packet is *innovative*. A packet is considered innovative if it is linearly independent from the previous packets of the same block that the destination has received. If the coded packet is not innovative, it will be discarded. Each coded packet represents a linear equation of the m native packets and the coding coefficients are known to the destination via the embedded code vector. Thus, as long as m innovative packets have been collected, the destination is able to recover the native packets. The decoding process in the destination involves solving the following set of linear equations, for example by Gaussian elimination algorithm. When the rank of the matrix is m, (i.e., there is no linear dependence between the m coded packets) the linear equations can be uniquely solved.

$$\begin{pmatrix} c_{11} & c_{12} & \cdots & c_{1m} \\ c_{21} & c_{22} & \cdots & c_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ c_{m1} & c_{m2} & \cdots & c_{mm} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{pmatrix} = \begin{pmatrix} x'_1 \\ x'_2 \\ \vdots \\ x'_m \end{pmatrix}$$

In these equations, an x'_i is a coded packet received by the destination, while its corresponding code vector is $\vec{c}_i = (c_{i1}, c_{i2}, \ldots, c_{im})$, and x_i-s are native packets.

The nodes in the sending cluster may have received more than one packet from the previous cluster, so that they can create its own combination. To generate a random linear combination, a node combines all the received packets of the same block with randomly selected coefficients.

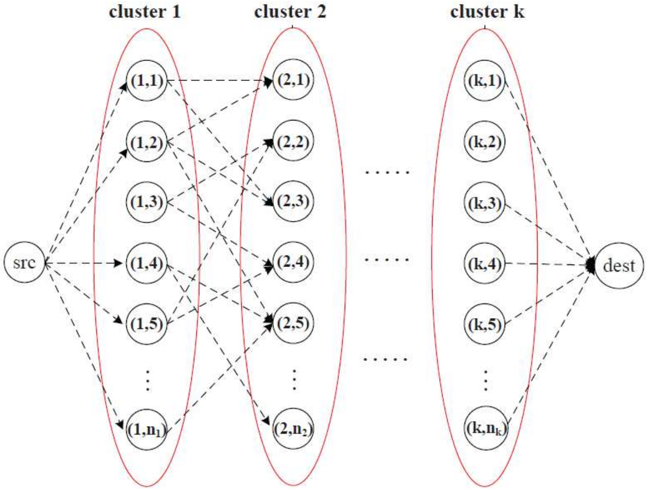


Figure 4: The cluster-to-cluster transmission model

B. *System Model*

Fig. 4 demonstrates our system model when a source node transmits packets to a destination. There are k clusters between the source and the destination nodes. Cluster i consists of n_i cooperating nodes which are relatively close to each other. Furthermore, the nodes in the same cluster are assumed to be within the transmission range of each other. The j^{th} node in cluster i is represented as node (i, j), where $i \in \{1, 2, \ldots, k\}$ and $j \in \{1, 2, \ldots, n_i\}$.

Two nodes from neighboring clusters may not necessarily be connected at all times. We define connectivity as the condition evaluated at a specific time that a node in the receiving cluster can receive a packet from a node in the sending cluster when the sending node has been scheduled to transmit the packet at the specific time. We define r_{ij}, associated with the node (i, j), as the number of nodes in the cluster $i + 1$ that are connected (i.e., can receive the transmission) from the node (i, j). For example, in Fig. 4, $r_{12} = 3$. Three nodes, the nodes (2, 1), (2, 3), and (2, 5), can receive transmission from the node (1, 2). Moreover, r_{kj} , $j \in \{1, 2 \ldots, n_k\}$, is an indication whether the node (k, j) is connected to the destination (the k^{th} cluster is the last cluster). r_{kj} can be either zero or one. Also, we denote r_s as the number of nodes in cluster 1 which are connected to the source node. In general, the parameters r_{ij} and r_s can change with time based on the CC scheduler.

As discussed in Section IV-A2, for any two nodes from adjacent clusters that are connected, a transmission loss probability is defined. This probability depends on the quality of the wireless link [21], and we denote, $p_{(i,j)(i+1,q)}$, $i \in \{1, 2 \ldots, k - 1\}$, $j \in \{1, 2 \ldots, n_i\}$, $q \in \{1, 2 \ldots, n_i+1\}$, as the loss probability of a transmission on the link between the node (i, j) and the node $(i + 1, q)$. These probabilities for the links between the source and the nodes in cluster 1 are defined as $p_{s(1,q)}$, $q \in \{1, 2 \ldots, n_1\}$. The corresponding probabilities of the links between the nodes in the last cluster and the destination are denoted by $p_{(k,q)d}$, $q \in \{1, 2 \ldots, n_k\}$. Table I lists the parameters used in the paper.

TABLE I. PARAMATERS

n_i	Number of nodes in a cluster i
k	Number of clusters between the source and destination
r_{ij}	Number of nodes in cluster $i+1$ that are connected to node (i,j)
r_s	Number of nodes in cluster 1 that are connected to the source node
$p_{(x)(y)}$	The loss probability of a transmission over a link between node x and node y
m	Number of native packets in a block

V. ANALYSIS OF THE NUMBER OF NODES IN A CLUSTER

Although a larger cluster results in better performance (i.e., larger end-to-end successful receipt probability), it also leads to more transmissions and, thus, larger overhead. Thus, there is a tradeoff between the performance and the number of cluster nodes.

The goal is to maximize the decoding probability at the destination node, while maintaining the traffic below some level. To this end, we compute the probability that all the m native packets in the same block can be decoded when the size of cluster i is n_i. In our analysis, we make the following Fundamental Assumption: *a packet received by the destination and transmitted by the node in cluster k is innovative with high probability; i.e., the first m packets received by the destination are with high probability linearly independent.* Therefore, we assume that as long as the destination has gathered m packets, the m native packets can be decoded. We will show in Section VII that this assumption is, indeed, justified.

The assumption is based on the observation in [22], where it has been shown that the probability that a coded packet is useful to another node is $1-(1/2^q)$, when each relay node has abundant buffer to store coded packets. Recall that 2^q is the size of the Galois Field. Typically, $1-(1/2^q)$ is very close to 1 for practical values of q. The selection of the value of q depends on the tradeoff between sufficient linear independence and other parameters such as the overhead of the packet header, the buffer size, and the ease of implementation. Usually, q is chosen to be 8 (i.e., one byte is required to encode a coefficient). Thus, the size of the Galois Field is $2^q = 2^8 = 256$.

We define V_{ij}, where $1 < i \le k$, $1 \le j \le n$, as the probability that node (i,j) hears at least one coded packet transmitted from the nodes in the cluster i-1. Also, V_{1j}, $1 \le j \le n_i$, is the probability that at least one coded packet sent from the source node is received by a node $(1,j)$. We assume that the network connectivity is uniformly distributed, meaning that the probability that a node in the cluster i is connected to the node $((i+1),j)$ is equal for any j. Moreover, a node $(1,j)$ in cluster 1 can successfully hear the packet from the source with probability $\frac{r_s}{n_1} \times (1 - p_{s(1,j)})$. Also, the probability that node (i,j) can successfully receive a packet from the node $(i\text{-}1,t)$ is given by: $\frac{r_{(i-1)t}}{n_i} \times (1 - p_{(i-1,t)(i,j)})$. Then, we obtain that

$$V_{ij} = \begin{cases} 1 - \left(1 - \dfrac{r_s \times (1 - p_{s(i,j)})}{n_i}\right)^{m'}, & i = 1 \\ 1 - \displaystyle\prod_{t=1}^{n_{i-1}} \left(1 - \dfrac{V_{(i-1)t} \times r_{(i-1)t} \times (1 - p_{(i-1,t)(i,j)})}{n_i}\right), & 1 < i \le k \end{cases}$$

where we recall that m' stands for the number of coded packets the source actually transmits to cluster 1. Therefore, V_{ij} can be iteratively computed from the above equation.

Next, let P_s denote the probability that at least m packets are received by the destination. Using the above formula for V_{ij}, we calculate the value of P_s. Based on our Fundamental Assumption, P_s also represents the probability that all m native packets can be decoded at the destination. With the above analysis, we can compute P_s for a given number of nodes in each cluster. In addition, the expected number of native packets decoded in a block of size m is given by mP_s.

VI. LINEAR DEPENDENCE OF CODED PACKETS

The destination could obtain a coded packet which is transmitted from the node in cluster k, and then try to conduct the decoding process to retrieve the m native packets as long as the m coded packets have been gathered. In Section V, we make the assumption in the analysis, that the first m coded packets received by the destination can be decoded to retrieve the m native packets. That is the first m coded packets are linearly independent. Such an assumption is based on the observation that with very high probability each coded packet once received by the destination is innovative with respect to the packets in the same block. However, there may still exist linear dependence among the first m coded packets, and the destination needs to keep receiving more coded packets until the rank of the decoding matrix reaches m. This depends on how the coded packets are generated from the random linear combinations by each node when they are forwarded. When a node has more received packets stored in its buffer, with large probability it generates a coded packet, which is innovative to the node in the next cluster. Therefore, if there are more nodes with more coded packets, there is a higher probability that the destination obtains an innovative coded packet.

Whether the destination could obtain m linearly independent coded packets by receiving the least number of packets, depends on several parameters, such as n, r, and p. Generally, for larger n and r and for smaller p, the probability increases that the destination can retrieve the m native packets from the first m coded packets. This is because more packets are successfully transmitted to the next cluster and hence could be used to generate the linearly independent packets. In Section

VII, we will show how these parameters affect the linear dependence.

In order to make the destination collect m linearly independent packets by receiving fewest packets, we modify CC into *Enhanced CC (ECC)* with an operation that checks the linear independence of the coded packets. In ECC, once a node generates the random linear combination from its stored packets, it first checks whether the combination is linearly independent from the coded packets which have been already transmitted by any other node in the same cluster. If not, it would attempt to regenerate a new random linear combination until an innovative coded packet is obtained. However, since the node has limited number of packets in its buffer or the previously transmitted coded packets from the nodes in the same cluster can span the whole linear space, it may not be able to generate an innovative packet. Therefore, the regenerating process would be conducted up to a certain number of trials. In ECC, checking the linear independence could be implemented in our cluster model, because all the nodes in the same cluster are close to each other. Hence, each node knows which coded packets have been already transmitted by the other nodes in the same cluster.

VII. PERFORMANCE EVALUATION

In this section, we report the results of our performance evaluation of the proposed CC protocol and of the comparison of the CC protocol with other schemes. These results were obtained by an extensive simulation. Furthermore, the simulation results validate our analysis in Section V.

In our simulations, we used 2^8 as the size of the Galois Field over which network coding operations are performed. We consider a network with k clusters and n nodes in each cluster. A homogeneous case, where $\forall\, x,\, y : p_{(x)(y)} = p$, is considered. All the simulation results were obtained by averaging 10, 000 runs. In each run, $m = 10$ native packets in one block are routed from the source node to the destination node. We compare the performance of the CC protocol with two other cases, CNC and NCNC, as defined below.

CNC (*Cooperative with Non-Coding*) is the scheme which implements cooperative communication, but without network coding. Therefore, in CNC, native packets are directly transmitted, and cooperation is achieved by a node in a cluster transmitting a packet, which has not been already transmitted by any other node in the same cluster. Since all the nodes in the same cluster are within the transmission range of one another, each node knows which native packets have been already transmitted by the other nodes in the same cluster. As in CC, each node in a cluster is allowed to forward one packet only. After all the native packets in the same block have already been transmitted by the other nodes, the next transmitting node randomly chooses from the buffered native packets that have not been transmitted more than once.

The NCNC (*Non-Cooperative with Non-Coding*) scheme forwards packets without cooperation and without coding. Thus NCNC is the traditional routing protocol, where packets are transmitted only to the predetermined next hop relay node, without cooperation. For each native packet in a block, the source chooses a relay node in each cluster and forwards the native packet along the chosen relay nodes. Therefore, as in CC and CNC, there are at most n transmissions of packets of the same block from each cluster.

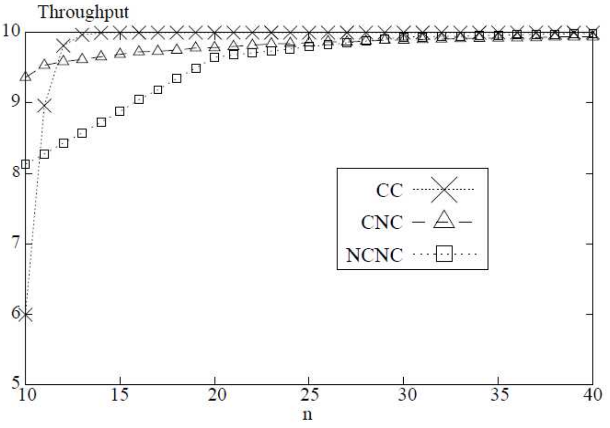
A. *The effect of n*

Fig. 5 shows how throughput varies with different number of nodes in a cluster. Throughput is defined as the total number of native packets which have been successful recovered at the destination from among the original $m = 10$ native packets. Fig. 5(b) shows results for $p = 0.1$, while Fig. 5(a) presents results for $p = 0.05$. In both cases, the throughput increases with n, since as there are more nodes in the cluster, it is more probably that a packet will reach the destination. Additionally, since in CC at least 10 coded packets are required for decoding, when the destination receives less than 10 packets, none of the packets could be recovered. Therefore, for n close to m, such as $n = 10$ the performance of the CC protocol is worse than the other schemes. However, for $n > 11$, the throughput in CC is higher than both of the other schemes and quickly reaches values close to 10, as in Fig. 5(a). This happens because transmitting more native packets in CNC only benefits the throughput if the same native packet is lost in a prior transmission. Otherwise, the transmission is a waste. On the other hand, with coding and with more transmissions, the CC scheme can recover the loss with very high probability. In contrast, NCNC which does not exploit cooperation and coding obtains the lowest throughput. Similar behavior is also observed in Fig. 5(b). Fig. 6 presents P_s, versus n. P_s defines the probability that all the 10 native packets can be decoded at the destination. The figure demonstrates that the smaller is the value of p or the larger is the value of n, the larger is P_s. The CC scheme significantly outperforms the CNC and the NCNC schemes in terms of P_s. Especially for $p = 0.05$ and $n \geq 13$, P_s of the CC scheme relatively quickly and closely approaches the value of 1.0. These results suggest that the proposed CC protocol can be applicable for error-sensitive applications.

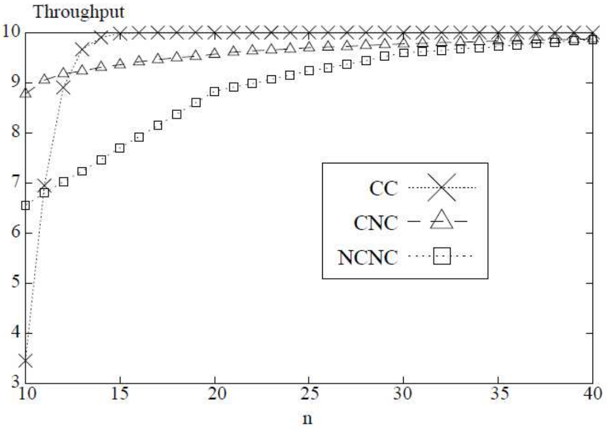
One of the major objectives of the CC scheme is to reduce the number of transmissions. Thus, in Fig. 7 we investigate the total number of transmissions in the network. The figure shows the number of transmissions which are required to achieve $P_s = 0.8$ for the different schemes. In this figure, p is set to 0.05. It is apparent from the results that the CC scheme needs the least number of transmissions. In other words, to achieve the same P_s, the CC protocol uses the smallest bandwidth among the three schemes.

B. *The effect of k*

We now further compare the performance of the schemes for different length (number of hops) of the route; i.e., different number of clusters between the source and the destination nodes. Fig. 8 and Fig. 9 illustrate how the throughput and P_s vary as k increases, respectively. For the results in these figures, each cluster consists of $n = 13$ nodes. The figures show that the CC scheme outperforms the CNC scheme due to the network coding operation, as it has been previously discussed. Besides, the performance of the CC and the CNC schemes is relatively constant as a function of k. The reason is that the nodes in the same cluster help each other to relay the packets by cooperation, making the cluster-to-cluster transmission more reliable. Therefore, despite longer route and more hops to the destination, information can still be forwarded reliably between clusters. However, due to lack of cooperation, the NCNC scheme is more prone to packet loses in each hop. Thus for longer routes, each hop has more significant effect on the overall end-to-end performance.



(a) $p = 0.05$



(b) $p = 0.1$

Figure 5: Comparison of throughput as a function of n

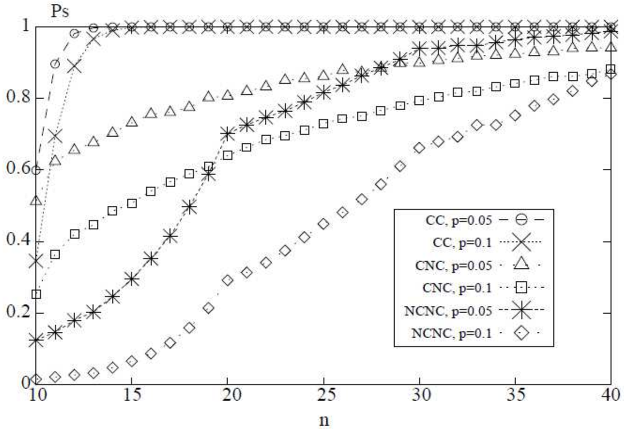


Figure 6: Comparison of P_s as a function of n

C. Validation of the analysis

Next, we validate the accuracy of our analysis of P_s. In Fig. 10, we plot P_s of the CC protocol as a function of n and p when r (including r_{ij} and r_s) is set 8. The analytical curve represents the numerical values computed from Section V. The simulation curve is obtained from the average of 10,000 independent runs. We observe that for different values of p, the analytical and the simulation results match nearly perfectly. Furthermore, the match of the simulation and the analytical results confirms the validity of our Fundamental Assumption. Finally, as already pointed out previously, lower p and larger n leads to larger P_s.

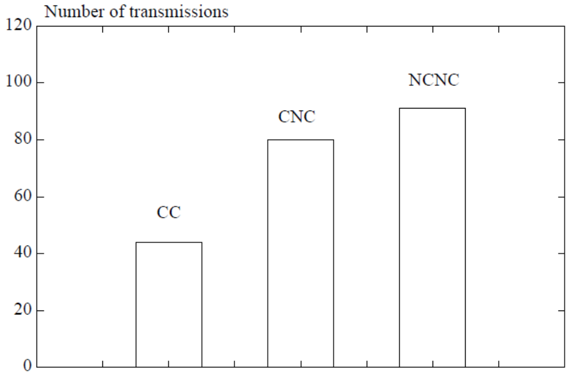


Figure 7: Comparison of total number of transmissions of the different schemes

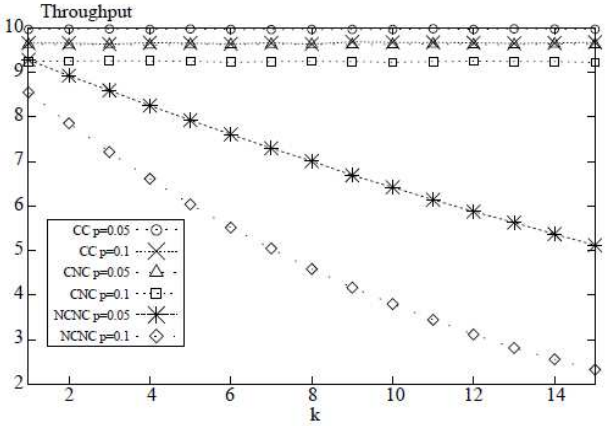


Figure 8: Comparison of throughput as a function of k

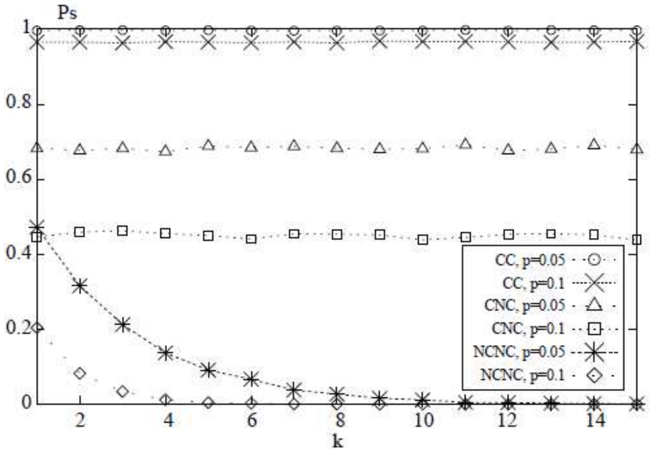


Figure 9: Comparison of P_s as a function of k

One of the main motivations for reducing the number of nodes in a cluster is to control the traffic. In Fig. 10, we can see that when n reaches a particular threshold, the improvement in P_s is only marginal. For example, for $p = 0.1$ and $n = 14$, the value of P_s is very close to 1.0. Thus, there is only a limited benefit in increasing n above the value of 14. In other words, the benefit of increasing the size of the cluster above this threshold is not worth the increased number of transmissions that would result from the larger clusters.

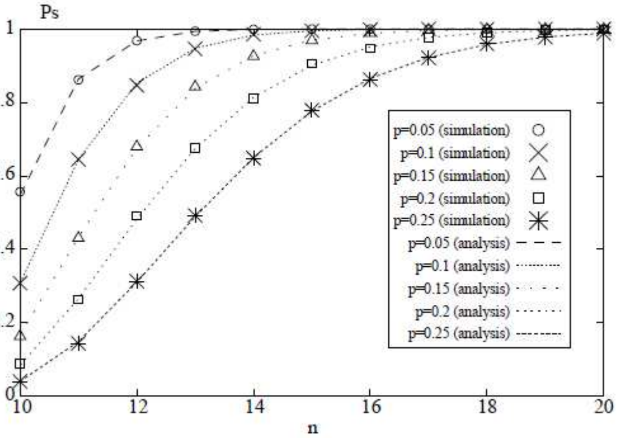


Figure 10: P_s vs. n for different p in CC

From the system design view, since more nodes in a cluster causes more traffic in the network, the clusters should be made as small as possible, given the level of required network performance, such as reliability or throughput. A good rule-of-thumb for p of 0.1 would be to keep the cluster size at about 15 nodes, which would result in the value of P_s close to 1.0.

VIII. CONCLUSTION

In this paper, we introduced the cluster-based Cooperative Communication scheme, which integrates cooperative communication with network coding. The basic idea behind the scheme is to exploit the cooperation to improve communication reliability and to leverage network coding to reduce the number of packet transmissions.

We analyzed the probability of successful reception of transmitted packets, and we showed how to optimize the number of nodes in the cluster, as to trade off between performance and overhead. We also derived a general rule-of-thumb that the size of the cluster should be kept at around 15 nodes. We compared the performance of the proposed scheme with schemes that do not incorporate cooperation or which do not incorporate network coding and we conclude that our scheme exhibit superior performance relative to the other simulated schemes.

IX. ACKNOWLEDGEMENTS

This work was supported by the NSF grant CNS-0626751 and by the AFOSR contract number FA9550-09-1-0121/Z806001.

X. REFERENCES

[1] A. Nosratinia, T.E. Hunter, and A. Hedayat, "Cooperative communication in wireless Networks," *IEEE Commun. Mag*., vol. 42, pp. 74–80, Oct. 2004.

[2] R. U. Nabar, H. Bolcskei, and F. W. Kneubuhler, "Fading relay channels: Performance limits and space-time signal design," *IEEE J. Sel. Areas Commun.*, vol. 22, no. 6, pp. 1099–1109, June 2004.

[3] J. N. Laneman, D. N. C. Tse, and G. W. Wornell, "Cooperative diversity in wireless networks: efficient protocols and outage behavior," I*EEE Trans. Inf. Theory*, vol. 50, no. 12, pp. 3062–3080, Dec. 2004.

[4] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, "Network information flow," *IEEE Trans. Inf. Theory*, vol. 46, no. 4, pp. 1204–1216, July 2000.

[5] A. Sendonaris, E. Erkip, and B. Aazhang, "User cooperation diversity - part I: system description," *IEEE Trans. Commun*., vol. 51, no. 11, pp.1927–1938, Nov. 2003.

[6] M. Gastpar and M. Vetterli, "On the capacity of large Gaussian relay networks," IEEE Trans. Inf. Theory, vol. 51, no. 3, pp. 765–779, Mar. 2005.

[7] S. Cui, A. J. Goldsmith, and A. Bahai, "Energy-efficiency of MIMO and cooperative MIMO techniques in sensor networks," *IEEE J. Sel. Areas Commun*., vol. 22, no. 6, pp. 1089–1098, Aug. 2004.

[8] L. S. Pillutla and V. Krishnamurthy, "Joint rate and cluster optimization in cooperative MIMO sensor networks," in *IEEE 6th Workshop on Signal Processing Advances in Wireless Communications*, Philadelphia, PA, USA, Mar. 2005, pp. 265–269.

[9] M. Elhawary and Z. J. Haas, "Energy efficient protocol for cooperative transmission," submit to publication.

[10] A. del Coso, U. Spagnolini, and C. Ibars, "Cooperative distributed MIMO channels in wireless sensor networks," *IEEE J. Sel. Areas Commun*., vol. 25, no. 2, pp. 402–414, Feb. 2007.

[11] S. Biswas and R. Morris, "ExOR: opportunistic multi-hop routing for wireless networks ," in *Proc. of ACM SIGCOMM*, Philadelphia, PA, USA, Aug. 2005, pp. 133–144.

[12] S. Chachulski, M. Jennings, S. Katti, and D. Katabi, "Trading structure for randomness in wireless opportunistic routing," in *Proc. of ACM SIGCOMM*, Kyoto, Japan, Aug. 2007, pp. 169–180.

[13] R. Koetter and M. Medard, "An algebraic approach to network coding," *IEEE/ACM Trans. Netw*., vol. 11, no. 5, pp. 782–795, Oct. 2003.

[14] S.-Y. R. Li, R. W. Yeung, and N. Cai, "Linear network coding," *IEEE Trans. Inf. Theory*, vol. 49, no. 2, pp. 371–381, Feb. 2003.

[15] D. S. Lun, N. Ratnakar, R. Koetter, M. Medard, E. Ahmed, and H. Lee, "Achieving minimum-cost multicast: a decentralized approach based on network coding," in *Proc. of IEEE INFOCOM*, Miami, FL, USA, Mar. 2005, pp. 1607–1617.

[16] C. Fragouli, J. Widmer, and J.-Y. L. Boudec, "A network coding approach to energy efficient broadcasting: from theory to practice," in *Proc. of IEEE INFOCOM*, Barcelona, Spain, Apr. 2006, pp. 1–11.

[17] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Medard, and J. Crowcroft, " XORs in the air: practical wireless network coding," in *Proc. of ACM SIGCOMM*, Pisa, Italy, Sept. 2006, pp. 243–254.

[18] S. Sengupta, S. Rayanchu, and S. Banerjee, "An analysis of wireless network coding for unicast sessions: the case for coding-aware routing," in *Proc. of IEEE INFOCOM*, Anchorage, AL, USA, May 2007, pp. 1028–1036.

[19] C. E. Perkins and E. M. Royer, "Ad hoc On-Demand distance vector routing," in *the 2nd IEEE Workshop on Mobile Computing Systems and Applications*, New Orleans, LA, USA, Feb. 1999, pp. 90–100.

[20] D. B. Johnson and D. A. Maltz, "Dynamic source routing in Ad Hoc wireless networks," in *Mobile Computing, edited by Tomasz Imielinski and Hank Korth*, chapter 5, Kluwer Academic Publishers, 1996, pp. 153–181.

[21] D. Aguayo, J. Bicket, S. Biswas, G. Judd, and R. Morris, "Link-level measurements from an 802.11b mesh networkg," in *Proc. of ACM SIGCOMM*, New York, NY, USA, 2004, pp. 121–132.

[22] S. Deb, M. Medard, and C. Choute, "Algebraic gossip: a networkcoding approach to optimal multiple rumor mongering," *IEEE Trans. Inf. Theory*, vol. 52, no. 6, pp. 2486–2507, June 2006.