Coded Cooperative Data Exchange for Multiple Unicasts

Shahriar Etemadi Tajbakhsh and Parastoo Sadeghi
Research School of Engineering
The Australian National University
Canberra, 0200, Australia
Emails: {shahriar.etemadi-tajbakhsh, parastoo.sadeghi}@anu.edu.au

***Abstract*—The advantages of coded cooperative data exchange has been studied in the literature. In this problem, a group of wireless clients are interested in the same set of packets (a multicast scenario). Each client initially holds a subset of packets and wills to obtain its missing packets in a cooperative setting by exchanging packets with its peers. Cooperation via short range transmission links among the clients (which are faster, cheaper and more reliable) is an alternative for retransmissions by the base station. In this paper, we extend the problem of cooperative data exchange to the case of multiple unicasts to a set of n clients, where each client c_i is interested in a specific message x_i and the clients cooperate with each others to compensate the errors occurred over the downlink. Moreover, our proposed method maintains the secrecy of individuals' messages at the price of a substantially small overhead.**

I. Introduction

Imagine that a group of nearby wireless clients are interested in downloading the same set of data packets from a base station (a multicast scenario). As a very common issue, some of the clients might miss some of the packets because erasures usually happen over wireless links due to fading, noise, etc. Conventionally, the base station would be in charge to retransmit the missing packets. However, a set of recent research papers proposed another solution: If the clients have received the set of packets collectively, they can cooperate by exchanging packets to obtain the entire set [1]–[6]. Moreover, in such a problem setting which is called the *coded cooperative data exchange* (CCDE) problem, network coding techniques have been shown to be substantially advantageous in terms of bandwidth efficiency and the energy consumed by the clients.

The main benefit of such a cooperation is that, a fraction of the bandwidth which should be allocated for retransmissions by the base station is freed and also short range transmission links among the clients are usually faster, cheaper and more reliable. Network coding reduces the total number of transmissions required to satisfy the demands of all the clients in this cooperative setting. Fig. 1 shows an example of cooperative data exchange. Assume three clients c_1, c_2 and c_3 have initially received the sets $\{x_1, x_2\}$, $\{x_2, x_3\}$ and $\{x_1, x_3\}$, respectively. In this example, using network coding, two transmissions are required to provide the missing packets to each client as shown in Fig. 1.

In this paper we extend the idea of coded cooperative data exchange (which was originally introduced for multicast applications) to the case of multiple unicasts. In this case, a group of wireless clients each client c_i is interested in a specific message x_i. In a conventional communication system, each client c_i might miss its own message x_i because of erasure. We provide a method to bring the advantages of cooperation via short range transmission links to the case of multiple unicasts where the clients cooperate to recover the missing information by each client. Unicasts form a large portion of communication networks traffic. Moreover, as a crucial consideration of the end users, our approach maintains the secrecy of the individuals' messages.

In the proposed method in this paper, different messages x_i, $i = 1, \ldots, n$ are combined together using a special coding process at the base station called partial random linear network coding (P-RLNC), resulting in a set of coded packets say $p_1, \ldots, p_n$, which are broadcast to all the clients. Then a private decoding vector of coefficients is provided to each client. Each client needs a subset of the coded packets (according to its decoding vector) to be able to decode its own message. However each client might have missed some of the packets required for decoding, so it needs to cooperate with its neighbors to obtain the missing packets and if nobody can provide that packet, it should be retransmitted by the base station.

Fig. 2 shows an example of our proposed approach. Suppose we want to deliver four messages x_1, x_2, x_3 and x_4 to four clients c_1, c_2, c_3 and c_4, respectively. To generate the coded packets p_1, p_2, p_3 and p_4, the base station solves the following system of equations, assuming all the operations are done over a finite field $\mathbb{F}_q$.

$$\begin{cases} x_1 = 2p_1 + 3p_3 \\ x_2 = 4p_2 + 5p_4 \\ x_3 = p_1 + p_4 \\ x_4 = 3p_2 + 4p_3 \end{cases}$$

Suppose clients c_1, c_2, c_3 and c_4 have initially received the subsets $P_1 = \{p_1, p_2, p_4\}$, $P_2 = \{p_1, p_2, p_3\}$, $P_3 = \{p_3, p_4\}$ and $P_4 = \{p_1, p_2, p_4\}$, respectively, due to erasures. The base station sends an exclusive and private decoding vector to each client (using a low overhead method which is discussed in Subsection III-B). For example, it sends the vector $[0, 3, 4, 0]$ to c_4. Based on the information in these vectors, each clients recognizes that which packets it needs to obtain its own

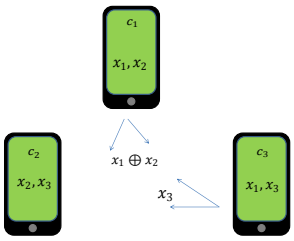

Fig. 1. Cooperative data exchange in a broadcast scenario.

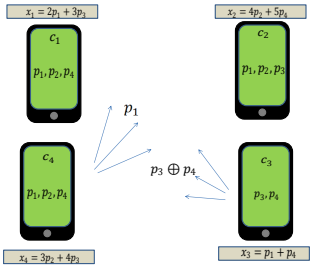

Fig. 2. Cooperative data exchange for multiple unicasts.

message. Then, each client informs the others about which packets it has already received and which packets it is looking for. Now assume all the clients share an error-free wireless broadcast channel. If client c_3 transmit the re-encoded packet $p_3 \oplus p_4$ and client c_4 only forwards the packet p_1, then each client would be able to obtain whatever it needs to decode its intended original message. It should be noted that client c_3 neither would be able to obtain packet p_2 nor needs it to decode the message x_3.

The rest of this paper is organized as follows: Section II reviews the available literature around the subject of this paper. Different phases of the proposed method are discussed in detail in Section III. In Section IV, two theorems are proven to show the security level and network coding gains of the proposed system. Section V numerically evaluates the performance of the method proposed in this paper and the paper is concluded in Section VI.

II. Related Work

The idea of cooperative data exchange has been studied in a set of recent research papers (sometimes under different names). [1] introduced the idea of cooperation for recovery purposes as an alternate for conventional retransmission mechanisms in a broadcast scenario. Network coding is applied to the same scenario in [2]–[5] where [3] provides some lower bounds and upper bounds on the total number of transmissions and a heuristic method is proposed to find the minimum number of transmissions. References [5], [6] provide a deeper understanding of the problem by introducing analytical network flow models to this problem. However, all of these works consider broadcast applications where all the clients are interested in the same content. In this paper our aim is to exploit the advantages of such cooperative data exchange settings for multiple unicast sessions where each client is interested in a different content. In [7], the capacity of broadcast channels with cooperating receivers is studied, and specifically the case that two receivers are interested in different messages but willing to cooperate is considered. Such a scenario is conceptually similar to our work.

Similar to many other cooperative and peer-to-peer network settings, security is a major concern of this paper. The relationship between network codes and security has been studied in many recent research papers. Information theoretic security as a strong measure for security level can be achieved even if a limited number of links are wiretapped by an adversary in a wired network multicast scenario [8]. Reference [9] shows that if the condition of information theoretic security is relaxed to *weak security*, higher secure data transmission rates would be achievable [8]. If transmission of information is weakly secure, it means the wiretap is not able to obtain any *meaningful* information. As an example, one can not decode the XOR combination of two bits, although it carries one bit of information, so in this case the coding is weakly secure. Reference [10] considers the case that instead of an external wiretapper, a group of nodes within the network are malicious. Unlike the mentioned works in the secure network coding literature, we assume that the corresponding vectors of network coding coefficients are privately provided to the clients individually. In other words, network coding coefficients play a crucial role as private keys.

III. System Model and Proposed Method

As discussed earlier in the example of Fig. 2, our main goal is to introduce a method based on a cooperative scheme among a group of n clients $C = \{c_1, \ldots, c_n\}$, where each client c_i is interested in secure reception of a distinct message $x_i \in X = \{x_1, \ldots, x_n\}$. Without loss of generality, each x_i is an element of a finite field $\mathbb{F}_q$. We denote the vector of all messages by $\mathbf{X} = [x_1, \ldots, x_n]$. By 'secure' we mean each client c_i is supposed to be able to obtain its own message x_i but not the other ones'.

Our proposed method includes three phases: (i) Broadcast (ii) Key sharing and (iii) Cooperation. In the broadcast phase, the messages are combined using a special kind of random linear network coding and the resulting packets are transmitted over a wireless broadcast channel to all the clients. The set of coded packets are denoted by $P = \{p_1, \ldots, p_n\}$ which are broadcast to all the clients. Each client c_i may receive only a subset of packets $P_i \subseteq P$. However, network coding coefficients are not collectively sent to all the clients, but each client receives only the decoding coefficients it requires to reveal its own message which is done using a low overhead algorithm in the key sharing phase. Each client needs to listen to the entire transmission from the base station because it does not know which coded packets are needed for decoding. This will make cooperation at a later stage more meaningful and feasible.

Each client might not have received all the packets required for decoding its own message. Assuming that all the clients are interested in cooperating with each other, each client can help its neighbors to obtain their intended messages. In the rest

of this section, we will discuss the mentioned three phases in more detail.

A. Broadcast Phase

To give a formal description of the broadcast phase, we define a decoding matrix $\mathbf{A}$, and we denote the i-th row of $\mathbf{A}$ by $\mathbf{A}_i$. At each row $\mathbf{A}_i$, the number of non-zero elements is denoted by r_i, where these elements are randomly drawn from the finite field $\mathbb{F}_q$, and are placed in r_i random places (entries) in $\mathbf{A}_i$. The other elements of $\mathbf{A}_i$ are set to be zero. The set of indices of non-zero elements of $\mathbf{A}_i$ is denoted by R_i. We defined our coding method in the broadcast phase as *partial random linear network coding* (P-RLNC) since a subset of coefficients are imposed by the algorithm to be zero (More details on generating matrix $\mathbf{A}$ is given in Subsection III-B). Therefore, each client needs only to receive a fraction of the entire set of packets to be able to decode its own message which makes the proposed method more resilient against packet erasures. In other words, each client c_i might not need to receive all its missing packets (if erasures happen) except for those ones in R_i.

The base station solves the following system of linear equations to generate the set of packets P. For simplicity and without loss of generality, we assume r_i is identical for all the clients and is equal to $r < n$.

$$\mathbf{X}^T = \mathbf{A}\mathbf{P}^T$$

where $\mathbf{P} = [p_1, \ldots, p_n]$ is the vector of all the elements in the set P. Therefore, each message x_i would be a random linear combination of r elements in P. Clearly, matrix A should be of full rank, otherwise the above system of equations is not solvable. Further discussion on the rank of such sparse matrices can be found in [11].

All the packets in P are broadcast to all the clients. However, the elements of $\mathbf{A}_i$ (the coding coefficients) are not piggy-backed in the header of the packets, but are privately provided to client c_i using a private key encryption system. Therefore, each client can only reveal its own message. *If each client knows the identity of the packets it needs to decode its message before the broadcast phase starts, it can switch off its receiver when the rest of packets are transmitted to save its battery, but this will decrease the amount of side information at the receivers and consequently results in a reduction in network coding.*

B. Key Sharing Phase

In an ideal scenario without any erasures, client c_i holds the entire set of packets P. Therefore, it can obtain all the messages if it knows the matrix $\mathbf{A}$. To keep the secrecy of messages, each row $\mathbf{A}_i$ should be privately delivered to the corresponding client c_i. This can be done separately for each client using any standard encryption system and a secret message be sent to each client. However, in this paper, we propose an encryption method for secret sharing of the decoding coefficients with a small transmission overhead. In particular, with a single pair of public messages the decoding coefficients are sent to all the clients (each client infers a different *meaning* from the same message as it is described here).

Definition 1. *We denote a permutation of the elements of a set B with $\pi_B^i : B \to B$, which is a one-to-one and covering function which (randomly) maps every element $\alpha \in B$ to another element in $\beta \in B$ ($\pi_B^i(\alpha) = \beta$). i is an arbitrary index which is used later to specify a client.*

Definition 2. *We denote the set of all non-zero elements in $\mathbb{F}_q$ by $Q = \{\alpha_1, \ldots, \alpha_{q-1}\}$. Also, $\tilde{N} = \{1, \ldots, n\}$ is the set of all positive integers smaller than $n + 1$.*

Here, the procedure for generating the decoding matrix $\mathbf{A}$ and sharing the decoding vector $\mathbf{A}_i$ is described:

1) The base station generates a random subset of r elements randomly drawn from Q denoted by Z_r and a subset of r elements from the set $\tilde{N}$ denoted by Y_r. The elements of the sets Y_r and Z_r are represented by y_i and z_i for $i = 1, \ldots, r$, respectively.
2) Each row $\mathbf{A}_i$ of the matrix of the decoding coefficients is generated by setting $\mathbf{A}_i(\pi_{\tilde{N}}^i(y_j)) = \pi_Q^i(z_j)$ for $j = 1, \ldots, r$. The sets Y_r and Z_r are publicly broadcast to all the clients. $\pi_{\tilde{N}}^i$ and π_Q^i are the corresponding pair of permutations for client c_i which are securely provided to client c_i as a private key.
3) Each client c_i can use the reverse functions of $\pi_{\tilde{N}}^i$ and π_Q^i to decrypt the set of decoding coefficients. Although the clients are receiving the same message from the base station, the decryption process in each client c_i results in different set of decoding coefficients as the functions $\pi_{\tilde{N}}^i$ and π_Q^i are different for each client.

Each client c_i can only decode its own message as it needs to access the functions $\pi_{\tilde{N}}^j$ and π_Q^j if it wants to eavesdrop and decode the packets of client c_j, $j \neq i$.

C. Cooperative Phase

Each client might have received only a subset of the packets due to packet erasures caused by fading, noise, etc. We assume the clients are willing to cooperate to obtain their own messages by exchanging some (coded) packets. We define a matrix S to denote the status of each packet for all clients. Each element s_{ij} in S takes one of the following three values depending on the status of packet p_j for client c_i:

- $s_{ij} = 1$ if client c_i initially holds packet p_j. The set of packets initially received by c_i is represented by Γ_i.
- $s_{ij} = 2$ if client c_i initially does not hold packet p_j but needs that packet to recover its own message. The set of such elements for client c_i is denoted by Ω_i.
- $s_{ij} = 3$ if client c_i neither holds the packet p_j nor needs it to recover its own message. These elements are shown by Ψ_i.

Unlike the scenario discussed in [3], [5], which can be modeled as a multicast with side information at the sinks and is solved using a linear programming, the problem introduced in this paper is a non-multicast problem in general which is

not easily solvable [12], [13]. The reason is that each client is not interested in the same set of packets as the other ones. If $\Psi_i = \emptyset$ for all the clients, the problem is reduced to the CCDE problem defined in [3], [5].

Here we provide a semi-distributed algorithm to find the best combination of packets that most number of recipients can instantaneously decode. The set of all clients who might be placed all over the coverage area of the base station can be divided to small clusters of nearby clients, where all the clients within are assumed to be in the radio range of each others. In this paper, we assume each client c_i can only cooperate with its peers within the same cluster by exchanging *re-encoded packets* to obtain the set of missing packets.

Definition 3. *We define the* satisfaction degree *of a set of packets $U_i^t \subseteq \Gamma_i$, $t = 1, \ldots, 2^{|\Gamma_i|}$ denoted by $d(U_i^t)$ as the total number of clients such as c_j that if each receives an XOR-ed version of all the packets in U_i^t, i.e. $\bigoplus_{\ell:p_\ell \in U_i^t} p_\ell$ it can decode and obtain a packet such as $p_j \in \Omega_i$. The index t denotes an arbitrary order to represent the set of all subsets of the set Γ_i.*

Clearly, such a combination is decodeable for client c_j if the client c_i holds all the packets in U_i^t except one. However, to be included in the satisfaction degree of a set U_i^t, the result of decoding process should be also useful which means client c_j obtains a packet in Ω_j. The details of cooperation algorithm is as follows.

1) Each client c_i transmits the vector $s_i = \{s_{i1}, \ldots, s_{in}\}$ to all its peers in the same cluster. So each client will know about the demands of the other clients.
2) Each client calculates the satisfaction degree of all subsets $U_i^t \subseteq \Gamma_i$ and finds the maximum of satisfaction degree over all the possible subsets. Each clients announces this number to its neighbors.
3) The client who can provide the largest satisfaction degree generates a XOR-ed version of all the packets in the corresponding set and broadcasts it to its neighbors.
4) The matrix S and the sets Γ_i, Ω_i and Ψ_i are updated.

For instance, here we provide the application of the above definitions for the example depicted in Fig. 2. The corresponding matrix S which can be called as the *packet distribution matrix* would be represented as follows for the example in Fig. 2:

$$\mathbf{S} = \begin{pmatrix} 1 & 1 & 2 & 1 \\ 1 & 1 & 1 & 2 \\ 2 & 3 & 1 & 1 \\ 1 & 1 & 2 & 1 \end{pmatrix}$$

As an example, the sets $\Gamma_3 = \{p_3, p_4\}$, $\Omega_3 = \{p_2\}$ and $\Psi_3 = \{p_1\}$ represent what initially c_3 has received, what it needs and what would not be helpful for c_3 if it were included in any transmission, respectively. The satisfaction degrees of the sets $U_3^4 = \{p_3, p_4\}$ and $U_1^8 = \{p_1, p_2, p_4\}$ are $d(U_3^4) = 3$ and $d(U_1^8) = 1$ which means that while the combination $p_3 \oplus p_4$ is instantaneously decodeable and useful for three clients, the combination $p_1 \oplus p_2 \oplus p_4$ is only desirable for one client (i.e. c_2).

IV. On the Security and Performance

One might be concerned about the brute force attacks where an adversary client c_i attempts to guess the coding coefficients of c_j by trying all the possible choices for $\mathbf{A}_i$. At each trial, c_i makes a guess about $\mathbf{A}_i$ and examines its guess to check if the decoded message has any interpretation. In practice an attacker may need to decode a sequence of messages at each trial to observe a meaningful word (message). Theorem 1 formulates the average number of guesses an adversary client should make to be able to obtain a meaningful message.

Theorem 1. *An eavesdropping client c_i needs to try at least $\frac{\min\{(q-1)^r, (q-1)!n!\}+1}{2}$ guesses on average, to make a correct guess about the coding coefficients $\mathbf{A}_j$ of the client c_j, $j \neq i$.*

Proof: Any client c_j with at least one missing packet needs to participate in the cooperative phase. Therefore, it should announce the vector $\mathbf{s}_j = [s_{j1}, \ldots, s_{jn}]$ to all its neighbors. Hence, the adversary client c_j is notified of the set of all non-zero elements in $\mathbf{A}_i$. Each non-zero entry in $\mathbf{A}_i$ can take $q-1$ different values from the finite field $\mathbb{F}_q$. Therefore, c_i needs to make guesses over a space of $(q-1)^r$ different possibilities. However, if $(q-1)!n! < (q-1)^r$, then the attacker would prefer to make guesses over the permutation functions π_N^i and π_Q^i with $n!$ and $(q-1)!$ possibilities, respectively (Therefore, the entire space of guesses would be $q!n!$ according to the basic counting principle).

Based on the above discussion, the problem is equivalent to the problem that M keys out of N keys can open a locked door, and it can been shown (using some combinatorics) that on average $\frac{N+1}{M+1}$ trials are required to find the first correct key without replacement of the keys. In our problem, there is only one correct key, therefore $M = 1$. Depending on which of the two values $(q-1)!n!$ or $(q-1)^r$ is larger, the attacker decides to guess either over permutation functions or decoding coefficients. Therefore, the average number of guesses is obtained by $\frac{\min\{(q-1)^r, (q-1)!n!\}+1}{2}$. ∎

Another important issue is the gain of network coding in the cooperative phase. As mentioned earlier, the scenario discussed in this paper is a non-multicast problem in general, which makes it difficult to provide an exact analytical solution. However, here we provide two simple bounds for the total number of required transmissions.

Theorem 2. *Let the total number of required transmissions in an instance of the cooperative setting discussed in Subsection III-C with the initial packet distribution S be denoted by T. Then,*

$$\max_i \Big\{ \Big| \bigcup_{j:s_{ij}=2} \{p_j\} \Big| \Big\} \leq T \leq \Big| \bigcup_{i,j:s_{ij}=2} \{p_j\} \Big| \qquad (1)$$

Proof: Suppose the client c_i requires $\bar{n}_i = |\Omega_i|$ packets. Obviously, $\bar{n}_i = |\bigcup_{j:s_{ij}=2} \{p_j\}|$. Therefore, client c_i needs to receive at least $\bar{n}_i$ equations to be able to obtain its own message. Consequently, the lower bound on the total number of transmissions is imposed by the client with maximum

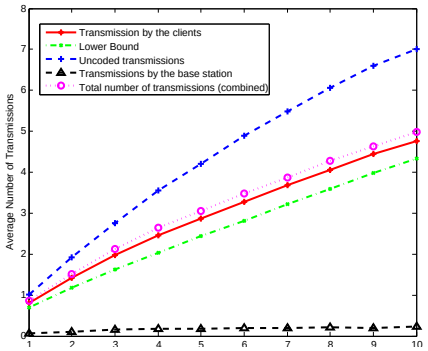

Fig. 3. Total number of transmissions in a multi-unicast scenario for different values of r.

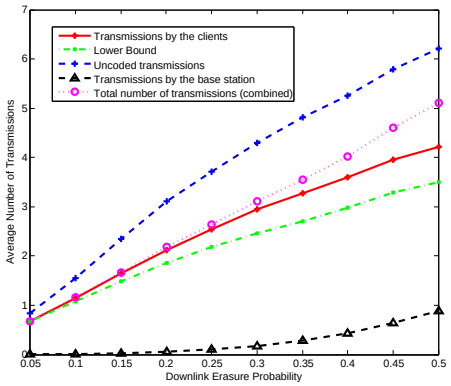

Fig. 4. Total number of transmissions in a multi-unicast scenario for different values of p.

$\bar{n}_i$. Moreover, the total number of required transmissions is trivially upper bounded by the size of the union of the sets of missing (but demanded by at least one client) packets by all the clients collectively which can be transmitted in an uncoded fashion. ■

V. Numerical Experiments

To examine the performance of the heuristic algorithm discussed in Section III, we have run some numerical experiments. One of the important parameters in our method is r, which is the number of coded packets that each user needs for decoding. As it can be inferred from the previous discussions, secrecy of the messages against computational attacks depends on r, where the larger the r, the more secrecy guaranteed. However smaller values for r is desirable in the sense that each client needs fewer packets to be able to decode its own message and consequently less packet exchange should be done among the clients. In other words a trade off between secrecy and throughput can be observed in our proposed system. The experiments have been run for a clusters of four clients out of $n = 10$ clients. Fig. 3 shows the average number of transmissions for different values of r which is compared with the bounds provided in Theorem 2. Also, as mentioned earlier, if a client needs a packet and would not be able to obtain it from its neighbors it should ask the base station to retransmit that packet. Fig. 3 also provides the average number of retransmissions by the base station as well as the average number of combined transmissions by the clients and the base station. The erasure probability for the downlink is assumed to be $p = 0.3$.

Fig. 4 relates to another experiment measuring similar items to Fig. 3 but against different values of downlink erasure probability and for a fixed value $r = 5$. In both figures we can see the considerable gain of network coding (almost 2) in comparison to uncoded transmissions.

VI. Conclusion

We extended the idea of coded cooperative data exchange which was originally proposed for multicast scenarios to the case of multiple unicast sessions to benefit from cooperation among the clients using short range transmission technologies. Moreover, our proposed method considers security issues to maintain a high level of secrecy for individual sessions.

Acknowledgment

This work was supported under Australian Research Council Discovery Projects funding scheme (project no. DP0984950 and project no. DP120100160).

References

[1] S. Raza, D. Li, C. N. Chuah, and G. Cheung, "Cooperative peer-to-peer repair for wireless multimedia broadcast," in *IEEE ICME*, Beijing, China, July. 2007.

[2] X. Liu, S. Raza, C. N. Chuah, and G. Cheung, "Network coding based cooperative peer-to-peer repair in wireless ad-hoc networks," in *IEEE International Conference of Communication (ICC)*, Beijing, China, May. 2008.

[3] S. E. Rouayheb, A. Sprinston, and P. Sadeghi, "On coding for cooperative data exchange," in *Proc. IEEE Inf.. Theory Workshop (ITW)*, Cairo, Egypt, Jan. 2010, pp. 118–122.

[4] A. Sprinston, P. Sadeghi, G. Booker, and S. E. Rouayheb, "A randomized algorithm and performance bounds for coded cooeprative data exchange," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Austin, TX, Jun. 2010, pp. 1888–1892.

[5] S. E. Tajbakhsh and P. Sadeghi, "A generalized model for cost and fairness analysis in coded cooperative data exchange," in *The 2011 International Symposium on Network Coding (Netcod)*, Beijing, China, July. 2011.

[6] T. Courtade, B. Xie, and R. Wesel, "Optimal exchange of packets for universal recovery in broadcast networks," in *Proc. IEEE Military Communications Conference (MILCOM)*, San Jose, CA, Nov. 2010.

[7] R. Dabora and S. D. Servetto, "Broadcast channels with cooperating decoders," *IEEE Trans. Inf. Theory*, vol. 52, pp. 5438 – 5454, 2006.

[8] N. Cai and R. W. Yeung, "Secure network coding," in *International Symposium on Information Theory (ISIT)*, 2002.

[9] K. Bhattad and K. R. Narayanan, "Weakly secure network coding," in *First Workshop on Network Coding, Theory, and Applications (NetCod)*, Italy, April 2005.

[10] L. Lima, M. Medard, and J. Barros, "Random linear network coding: A free cipher?" in *International Symposium on Information Theory (ISIT)*, Ithaca, NY, December 2007.

[11] G. Ma, Y. Xu, M. Lin, and Y. Xuan, "A content distribution system based on sparse linear network coding," in *IEEE Information Theory and Applications Workshop*, 2007.

[12] M. Medard, M. Effros, D. Karger, and T. Ho, "On coding for non-multicast networks," in *In Proceedings of 41st Annual Allerton Conference on Communication, Control, and Computing*, 2003.

[13] R. Dougherty, C. Freiling, and K. Zeger, "Insufficiency of linear coding in network information flow," *IEEE Trans. Inf. Theory*, vol. 51, pp. 2745 – 2759, 2005.