A Unified Channel-Network Coding Treatment for User Cooperation in Wireless Ad-Hoc Networks

Xingkai Bao and Jing Li (Tiffany)
Department of Electrical and Computer Engineering, Lehigh University, Bethlehem, PA 18015
Email: {xib3, jingli}@ece.lehigh.edu

Abstract— **We propose a combined channel-network coding solution for efficient user cooperation in wireless ad-hoc networks that comprise a host of terminals communicating to a common destination. The proposed framework, termed generalized adaptive network coded cooperation or GANCC, addresses the challenge of inter-user outage, which widely persists in practical cooperation scenarios, by adaptively matching code graphs to instantaneous network graphs (topologies). Additionally, GANCC treats channel codes as an integral part of the network code, and in doing so not only extracts the most benefit from these codes but also provides a live example supporting the notion that network codes are generalization of channel codes (as well as source codes).**

I. INTRODUCTION

Modern communication networks promise unprecedented capacity over conventional point-to-point communication links. Inherent to network communication is the cooperation among different users which pulls together all dimensions of communication resources[1]-[4]. User cooperation may occur in different forms, among different numbers of users, and in different layers of the network protocol stack. In the physical layer, user cooperation rooted back to the classic problem of relay channel in the seventies, and has recently evolved to the notion of cooperative diversity in the wireless context. In the network layer, the focus has been on cooperated routing and resource management. User cooperation is particularly beneficial for wireless systems, since while an individual channel operating alone may be useless due to severe path loss or deep channel fading, combined together a set of channels may become useful again.

We consider efficient cooperative strategies for wireless ad-hoc networks that comprise a host of users communicating with a common destination. We take a cross-layer approach and leverage the technologies from both the physical layer and the network layer. Of particular interest here is the joint treatment of channel coding and network coding to combat fading, the dominant channel impairment in the wireless environment.

Whereas channel coding has long been established as a fundamental technology for protecting bit streams from being corrupted by noise and fading, network coding has only recently found its way here. The technology, originated from the network flow problem, is a generalization of the traditional replicate-and-forward routing. By allowing intermediate relaying nodes to perform simple coding operation, network coding provides new capabilities to routing, and opens the possibility to achieve optimal throughput in *lossless* networks [5]-[6]. The application of network coding in *lossy* networks and particularly *wireless* networks only occurred in the last couple of years, but its potential to increase the diversity order and reduce the outage probability (in addition to improving the bandwidth efficiency) is already evident in [8]-[9].

Among the existing studies that exploit network coding in user cooperation, one work that proposes *adaptive network coded cooperation* (ANCC) is particularly noteworthy [8]. Unlike other approaches that use fixed network coding schemes and therefore rely on the ideal assumption of lossless inter-user channels [9], ANCC adaptively generates network codes on the fly by matching the code graph of some low density parity check (LDPC) code with the network graph that specifies the instantaneous network topologies. In this, ANCC has provided a practical and efficient solution to the changing and instable nature of wireless links and network topology, a concern that had previously prevented the deployment of network coding in wireless scenarios.

The ANCC protocol assumes that channel coding is performed separately from routing at the edge of the network. This assumption, although seemingly convenient, is in fact unnecessary as well as suboptimal. Discarding this assumption, here we generalize the ANCC protocol by considering a combined and unified treatment of channel coding and network coding. The idea finds its motivation and theoretic support in the emerging network information theory. Well-known from the classic Shannon information theory is the source-channel separation for *channels*, which states that source coding and channel coding over a communication channel can be performed independently from each other without loss of optimality. Recent studies[7] indicate that source-channel separation may also hold for *networks*, but source-network separation and channel-network separation will break. Hence, although source coding and channel coding may still be treated separately in such network scenarios as multiple access channels and broadcast channels, separating routing from source or channel coding will fail to bring the end-to-end optimality.

Rather than simply concatenating channel codes with network codes, the new protocol treats channel coding as an integral part of network coding. Following the notion developed in [7] that network codes are essentially generalization of source codes or channel codes, we refer to the new protocol as the *generalized adaptive network coded cooperation* (GANCC) protocol. We show that GANCC subsumes ANCC as its degenerated case. We also show that while ANCC has a

---

This project is supported by NSF under Grant No CCF-0430634, and by the Commonwealth of Pennsylvania, Department of Community and Economic Development, through PITA.

1-4244-0504-1/06/$20.00 ©2006 IEEE

network codeword length in $O(m)$,the number of cooperating users, the effective network codeword length of GANCC is in $O(Nm)$, the combined packet lengths from all the terminals (assuming all the packets have equal length N). This is achieved for GANCC even when each packet contains only N uncoded (and hence independent) data bits (i.e. no channel code for each packet). Hence, GANCC requires significantly fewer users to cooperate than ANCC to achieve a similar (network) coding gain.

Additionally, unlike any separate channel-network coding treatment where the rate allocation between the network code and the channel code needs to be carefully designed, with GANCC, the channel code and the network code are seamlessly integrated in one single codeword, with both functions merged and satisfied cohesively.

We begin in Section II with a brief introduction to ANCC, upon which GANCC is developed. The key idea of GANCC is demonstrated in Section III through a simple example, where no explicit channel codes are used in the source-packets. The general framework that works for both (channel) coded and uncoded source-packets is discussed in Section IV. Concluding remarks are provided in Section V.

II. ANCC

The model of interest here comprises m terminals communicating wirelessly to a common destination via two-phase user cooperation. In each phase, the m terminals transmit binary phase-shift keying (BPSK) modulated data through time division multiple access (TDMA).

We assume that all the communication channels used in this paper are spatially independent. Without loss of generality, we consider that each channel follows a frequency nonselective slow fading model with channel fading α and addictive noise Z. The fading coefficient α is modeled as a zero-mean, independent, circularly symmetric complex Gaussian random variable with unit variance, whose magnitudes $|\alpha|$ is Rayleigh distributed. Since user cooperation is most useful in time-limited channels, we consider the case where α remains constant during one round of user cooperation, and changes independently from one round to another. The channel noise Z captures the addictive channel noise and interference, and is modeled as a complex Gaussian random variable with zero mean and variance N_0.

The ANCC protocol proposed in [8] proceeds as follows. In the first phase, each terminal broadcasts its data packet of length N (referred to as source-packet) in its designated time slot. The terminals that are not transmitting listen and try to decode what it hears. Due to channel fading and other impairments, a terminal may not be able to retrieve all other source packets. We use *receive-set*, $\mathcal{R}(i)$, to denote the set of packets that Terminal i decoded correctly, where $\mathcal{R}(i) \subset \{1, 2, \cdots, m\}$.

In the second phase, each terminal randomly selects a small number of packets from its receive-set, computes their check-sum (i.e. adds those packets together symbol by symbol in the binary domain), and forwards the length-N check-sum packet (referred to as relay-packet) to the destination in its designated time slot. Meanwhile, those terminals that have not yet had a chance to relay continue to listen and decode. The correctly retrieved relay-packets will continue to be included in the decode-set $\mathcal{R}(i)$. Since the system operates in a TDMA manner, the decode-set satisfies $\mathcal{R}(i) \subset \{1, 2, \cdots, m, m+1, m+2, \cdots, m+i-1\}$, where $m+j$ denotes the relay-packet by Terminal j $(1 \le j < i)$. Hence, by the end of the second phase, the m terminals have transmitted, through user cooperation, a $(2m, m)$ network code in the form of a random, systematic, low-triangular low-density parity-check (LDPC) code. The source-packets transmitted in the first phase constitute the systematic symbols of the network code, and the relay-packets transmitted in the second phase constitute the parity symbols.

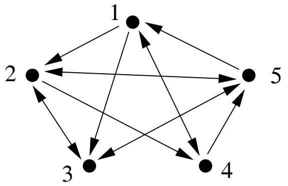

Fig. 1. An example of 5 users sending data to a common destination.

To illustrate, consider a simple example of $m = 5$ users. Assume that for a particular round of cooperation, the inter-user channels form an instantaneous network topology as shown in Fig. 1, where a directed link represents a quality connection that lasts throughout this round of cooperation (destination not shown in the figure). The receive-set of each user contains, respectively,

$$\begin{aligned} \mathcal{R}(1) &= \{\mathbf{1}, \mathbf{4}, \mathbf{5}\}, \\ \mathcal{R}(2) &= \{1, \mathbf{2}, \mathbf{3}, \mathbf{5}, \mathbf{6}\}, \\ \mathcal{R}(3) &= \{\mathbf{1}, 2, \mathbf{3}, \mathbf{5}, 6, \mathbf{7}\}, \\ \mathcal{R}(4) &= \{\mathbf{1}, \mathbf{2}, \mathbf{4}, \mathbf{6}, 7\}, \\ \mathcal{R}(5) &= \{\mathbf{2}, \mathbf{3}, \mathbf{4}, 5, \mathbf{7}, \mathbf{8}, \mathbf{9}\}. \end{aligned}$$

Assuming that the packets marked in bold font are selected (randomly) by each terminal to form check sums, add the selected packets together, we obtain a parity check matrix of the resulting network code:

$$H_{ancc} = \begin{array}{c} \begin{array}{cccccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \end{array} \\ \underbrace{\left[\begin{array}{ccccc} \mathbf{1} & 0 & 0 & \mathbf{1} & \mathbf{1} \\ 0 & \mathbf{1} & \mathbf{1} & 0 & \mathbf{1} \\ \mathbf{1} & 0 & \mathbf{1} & 0 & \mathbf{1} \\ \mathbf{1} & \mathbf{1} & 0 & \mathbf{1} & 0 \\ 0 & \mathbf{1} & \mathbf{1} & \mathbf{1} & 0 \end{array}\right.}_{\text{systematic symbols}} \underbrace{\left.\begin{array}{ccccc} 1 & 0 & 0 & 0 & 0 \\ \mathbf{1} & 1 & 0 & 0 & 0 \\ 0 & \mathbf{1} & 1 & 0 & 0 \\ \mathbf{1} & 0 & 0 & 1 & 0 \\ 0 & \mathbf{1} & \mathbf{1} & \mathbf{1} & 1 \end{array}\right]}_{\text{parity symbols}} \end{array} \tag{1}$$

Due to the random formation of the code, a small bit-map field needs to be included in the relay-packet, so that the destination knows how the checks are constructed and can correspondingly replicate the code graph and perform message passing decoding. Since a different network code is transmitted each round of user cooperation, an adaptive decoder architecture in the form of, for example, software-defined radio (SDR), needs to be implemented at the destination.

Depending on the quality of the user-destination channels or the residual power supply, a terminal may choose to relay multiple times, each time using a different relay-packet, or not to relay at all. The exploitation of user diversity and resource management in ANCC can bring additional cooperative benefits. Further, if there exists a simple feedback mechanism from the destination, and if the decoder complexity at the destination is not a concern, then the terminals can take continual turns to relay and stop as soon as the destination manages to successfully decode all the source-packets. The resulting network code has thus migrated itself from a fixed-rate LDPC code to a rateless digital fountain code [11].

III. GANCC: A SIMPLE EXEMPLARY CASE

A. Code Structure

The ANCC protocol does not consider or exploit the channel code which may well exist in each source-packet. Since the network code length is solely dependent on the number of users m, it takes a large number of users to cooperate in order to achieve a good network coding gain. The associated delay and management overhead can be costly. Further, in the case when a large cluster of co-located users are not possible (e.g. in a mobile ad-hoc network or a small-scale network), the network code length may be too small to provide a desirable coding gain.

The proposed GANCC protocol provides a remedy to this problem by integrating the channel codes and the network code in one single codeword, resulting in an effective code length of $2mN$, where N is the length of each packet (assuming the relaying phase takes the same time interval as the broadcasting phase). The beauty of GANCC is that the channel codes now constitute an *integral* part of the network code, rather than being loosely connected to the network code via iterative channel-network decoding. To best illustrate this, consider the extreme case where each source-packet contains only N *uncoded* raw bits with no explicit channel coding.

For simplicity, we consider the same 5-user example discussed in the previous section. The LDPC network code of ANCC, whose parity check matrix H_{ANCC} is given in (1), is rather weak due to the short block size (and the existence of length-4 cycles). The lack of channel coding in each packet further eliminates the possibility to iteratively decode the network code and the channel code to improve performance.

Now GANCC drastically changes the situation by a simple operation of *interleaving*: for each terminal, after selecting the packets from its receive-set, instead of computing their check-sums bit-by-bit in their original bit orders, the length-N bit-steams in the packets will first be interleaved, each using a different length-N scrambling pattern, before being added together for parities. Formally, the new parity check matrix, H_{GANCC}, is constructed by substituting each entry in H_{ANCC} with an $N \times N$ square matrix, where "0"s are replaced by null matrices, and "1"s are replaced by independent permutation matrices except for the "1"s on the right diagonal which are replaced by identity matrices (i.e. trivial permutations). The parity-check matrix H_{GANCC} that corresponds to H_{ANCC} in (1) is demonstrated in Fig. 2. In this figure, each permutation matrix, $\pi_{i,j}$, is a (random) row permutation of an identity matrix, whose row permutation pattern determines how User i scrambles User j's bit-stream. In the extreme but undesirable case where all the permutation matrices use the identity matrix, then GANCC degenerates to ANCC.

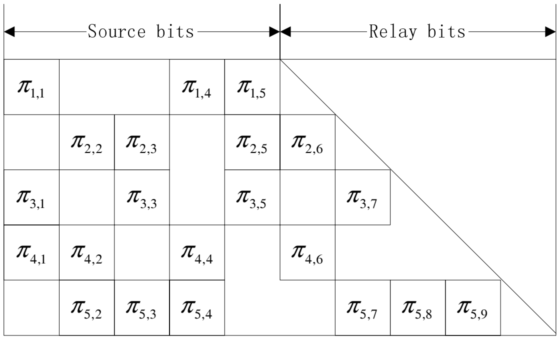

Fig. 2. An example of the parity check matrix for the unified channel-network code used in GANCC with uncoded source-packets ($m = 5$ terminals).

The permutation matrices or the interleavers are critical to the system performance of GANCC. First, interleaving integrates the bit-streams of all the users in one big network code in such a way that, although the bits in each bit-stream are uncoded and by themselves provide no inference about each other, interleaved and combined together they form an elegant "network" through which any one bit in any one bit-stream now carries information about the other. This integration brings an effective code length of $O(mN)$, where N typically ranges from a few hundred to a few thousand in practical systems. It therefore obviates the need for many terminals to coordinate and cooperate, making GANCC more practical. Second, by permuting each bit-stream using a different pattern, and so breaking the length-4 cycles that may previously exist in H_{ANCC}, interleaving reduces the chances for short cycles in H_{GANCC}. In the example, H_{ANCC} in (1) consists of several length-4 cycles, but the corresponding H_{GANCC} in Fig. 2 will have a much lower fraction of length-4 cycles if any.

Now to perform GANCC, each terminal needs to store a set of (random) interleaver, whose knowledge is also to be conveyed to the common destination. This consumes a large storage space on all the parties involved as well as a good amount of signaling overhead. To address this challenge, algebraic interleavers can be used in lieu of random interleavers. An algebraic interleavers is one whose scrambling pattern can be generated on-the-fly using an often-recursive formula with a few seeding parameters. Through the proper choice of formula and parameters, an algebraic interleaver can be made to behave much like a random interleaver, but requires significantly less storage.

Here, for the purpose of GANCC, our study reveals a solution that is even simpler than algebraic interleavers. The solution makes essential use of the recent advances in quasi-cyclic LDPC codes and particularly circulant LDPC codes [10], where it is found that circulant matrices/interleavers are efficient for constructing LDPC codes as well as lead to simpler encoding/decoding implementations. Hence, instead of using $N \times N$ random permutation matrices, we replace the "1"s in H_{ANCC} with $N \times N$ circulant matrices, such as the one in the below ($N = 5$):

$$\pi_{i,j} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix}. \qquad (2)$$

Since each row is the right cyclic shift of the previous row, it takes a single parameter, the position of the non-zero entry in the first row (denoted as p), to determine a circulant matrix. In practice, it is possible to make p a function of the terminals' indexes, thus eliminating any storage space and signaling overhead. For example, in a common situation where $N > 2m^2$, one can specify $(1 \le i \le m, 1 \le j < 2m)$

$$p(\pi_{i,j}) = 2m(i-1) + j + 1, \qquad (3)$$

which ensures no two circulant matrices repeat each other.

Clearly, GANCC now involves very little additional complexity than ANCC, yet produces a random LDPC (network) code with magnitudes larger of code length. We note that H_{GANCC} is random and tends to have a lower density than H_{ANCC}. In the example, H_{ANCC} in (1) appears rather dense, whereas the corresponding H_{GANCC} in Fig. 2 appears to have just the right density. In practice, a delicate balance needs to be accounted for when choosing the check degrees, since heavy density breaks the message-passing decoding and excessive sparsity leads to uselessly weak codes.

B. Experimental Results

To verify the efficiency of the proposed protocol, we present in Fig. 3 the simulated performance of GANCC using both random interleavers and circulant interleavers. For comparison purpose, the performance of ANCC (no interleaving) is also provided. We consider $m = 5$ users transmitting uncoded packets of length $N = 2000$. The network code has rate 1/2 and a message-passing algorithm with 30 decoding iterations is performed at the destination. We plot both the bit error rate (BER), averaged over all the bits in all the user packets (dashed curves), and the packet error rate (PER, solid curves), averaged over all the users, versus the user-destination signal-to-noise ratio (SNR, $Es/N0$). Since a different code is constructed every time, the curves represent the ensemble performance rather than that of a single code. That GANCC exhibits 7 dB gain in BER and 11 dB gain in PER than ANCC clearly points out the importance of interleaving and the large code length that comes after. Since the curves of random permutation matrices and circulant matrices hug together with no differentiable gap, it is thus safe to employ the simple circulant interleavers instead of random interleavers.

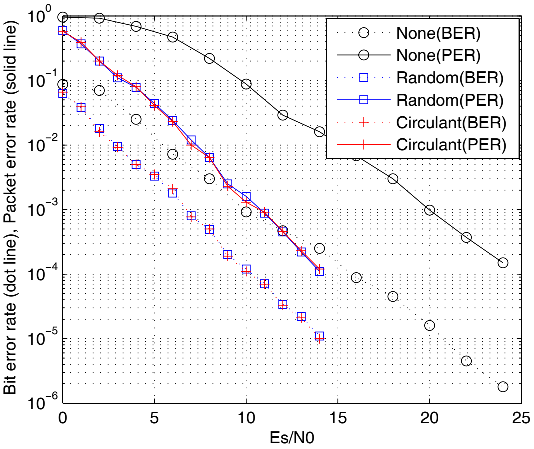

Fig. 3. The performance of GANCC with no explicit channel codes. $m = 5$, $N = 1000$.

IV. GANCC: THE GENERAL FRAMEWORK

Having discussed the key idea of GANCC using a simple example, here we present the general framework that unifies channel coding and network coding.

Assume all the packets consume the same bandwidth N (bits). Let $H_1, H_2, \cdots, H_m$ be the parity check matrices of the (N, K_1), (N, K_2), $\cdots$, (N, K_m) channel code used in each user packet, respectively, where K_i is the raw data size for each User i, and N is the length of the channel code.

The terminals distributively encode the network code using the same procedure as we described in the previous section, regardless of whether each packet is channel coded or not. That is, after broadcasting its own source-packet and collecting a decode-set, each terminal randomly selects a few packets from its decode-set, interleaves them using a different length-N circulant interleaver for each, adds them together bit-by-bit in the binary domain to form N parities, and forwards the length-N relay-packet to the destination.

Viewed from the destination, the combination of all the source-packets and the relay-packets together form one big network code whose parity check matrix consists of $2mN$ columns, pertaining to $\sum_i K_i$ raw data bits, $mN - \sum_i K_i$ "channel-parity" bits, and mN "network-parity" bits, and $2mN - \sum_i K_i$ rows, pertaining to $\sum_i (N - K_i)$ "channel-checks" and mN "network-checks". Considering the same 5-user example used in the previous sections, the parity check matrix H_{GANCC} of the network code will take the form in Fig. 4, where $\pi_{i,j}$'s are the (circulant) permutation matrices used in the network code, and H_i is the parity check matrix of the channel code used in the ith source-packet. When Terminal i does not employ a channel code, H_i becomes an identity matrix; and when all the channel codes degenerate, H_{GANCC} becomes the one in Fig. 2.

The unified channel-network code model depicted in Fig. 4 is general. It holds regardless of whether none, some, or all source-packets are coded, and regardless of what channel codes are used in those source-packets. (Although people seldom mention the parity check matrices of convolutional or turbo codes, they do exist as with any linear channel code.)

In terms of decoding, two strategies are available. The optimal decoder treats the H_{GANCC} as one single code and performs joint channel-network decoding at once. This is possible in practice when, for example, all the channel codes involved are individually suitable for message-passing decoding, and so will be the entire channel-network code. Alternatively, a two-level decoding architecture can be employed, such that the network code pertaining to the lower mN rows of H_{GANCC} in Fig. 4 is first decoded using the message-passing algorithm, whose soft (probabilistic) outcomes are then passed to the individual channel codes for channel decoding. If complexity permits and if all the channel codes produce soft reliability information, this soft information may iterate back to the network code for successive refinement, enabling iterative decoding between the channel code and the network code.

We comment that sequential decoding of the network code followed by the channel codes is what comes natural for ANCC. Iterative treatment between the network code and the channel codes is also a foreseeable extension for ANCC. However, joint (message-passing) decoding on one unified code graph will only emerge when we think of channel codes as an integral part of the network code, or think of network codes as generalization of channel codes. This is what GANCC is emphasizing.

Not to overstate the importance of this philosophy, we demonstrate in Fig. 5 the difference it can make. The simulation setup consists of $m = 5$ users, each employing a $(3, 6)$-regular LDPC channel code with length $N = 2000$.

The curves marked with “□” are the results from the joint channel-network decoding with 30 message-passing iterations, and those marked with “+” are from the separate decoding with 30 iterations for the network code and each individual channel code. With a similar complexity, the joint treatment evidently outperforms the separate treatment by 2 dB in average bit error rate and 3 dB in average packet error rate.

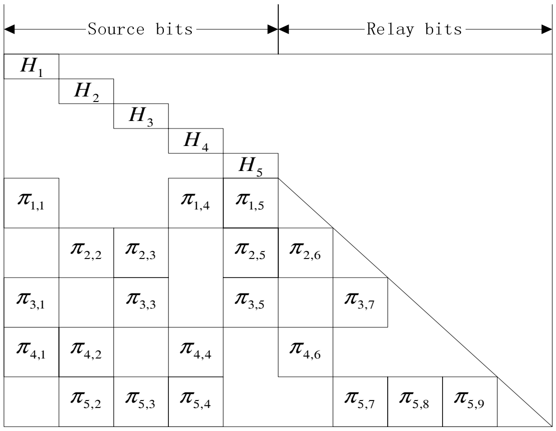

Fig. 4. An example of the parity check matrix for the unified channel-network code used in GANCC with channel-coded source-packets ($m = 5$ terminals).

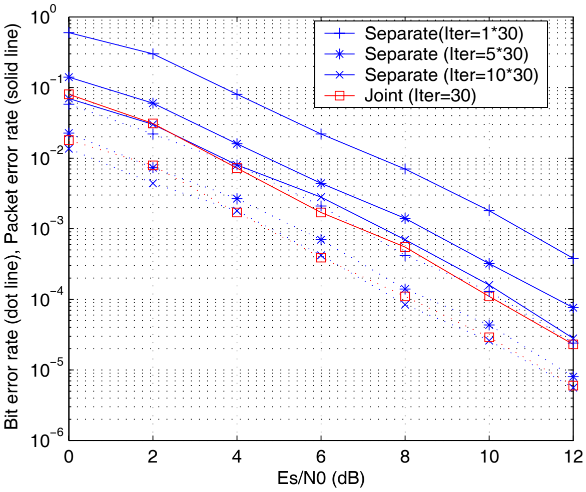

Fig. 5. Comparison of GANCC employing joint channel-network decoding and separate decoding. ($m = 5$ users)

V. CONCLUSIONS

To address the practical challenge of inter-user outage in wireless user cooperation, the adaptive network coded cooperation protocol proposed in [8] adaptively matches code graphs that specify random LDPC codes to network graphs that characterize instantaneous network topologies. This paper advances the protocol one step forward by integrating channel codes, which may well exist in the user packets, into the network code. The central theme of the proposed generalized adaptive network coded cooperation protocol, is to treat channel codes as an integral part of the network code and to view network codes are generalization of channel codes. In this, GANCC not only extracts the most benefit from both codes, but also makes the practice simpler and more practical, since a few users now suffice to yield a large coding gain.

REFERENCES

[1] T. E. Hunter and A. Nosratinia, “Coded Cooperation under slow fading, fast fading and power control,” *Proc. Asilomar Conf. Signals, Syst., Comput.,* 2002.

[2] M. Janani, A. Hedayat, T. Hunter, and A. Nosratinia, “Coded cooperation in wireless communications: space-time transmission and iterative decoding,” *IEEE Trans. Sig. Processing,* pp. 362-371, Febm 2004.

[3] J. N. Laneman, G. W. Wornell, “Distributed Space-Time-Coded protocols for exploiting cooperative diversity in wireless networks,” *IEEE Trans. Inform. Theory,* vol. 49, NO. 10, Oct 2003, pp. 2415-2425.

[4] G. Kramer, M. Gastpar, and P. Gupta, “Cooperative strategies and capacity theorems for relay networks,” to apper *IEEE Trans. Inform. Theory*, 2006.

[5] R. Ahlswede, N. Cai, S.-Y.R. Li and R.W. Yeung, “Network information flow”, IEEE-IT, vol. 46, pp. 1204-1216, 2000.

[6] R. Koetter, and M. Medard, “An algebraic approach to network coding,” *IEEE Trans. Networking*, Oct. 2003.

[7] M. Effros, M. Médard, T. HO, S. Ray, D. karger, and R. Koetter, *“Linear network codes: a unified framework for source, channel and network coding,”*, DIMACS workshop on network information theory, 2003.

[8] X. Bao, and J. Li, “Matching Code-on-Graph with Network-on-Graph: Adaptive Network Coding for Wireless Relay Networks,” *Proc. Allerton Conf. on Commun., Control and Computing* IL, Sept. 2005.

[9] C. Hausl, F. Schreckenbach and I. Oikonomidis, “Iterative network and channel decoding on a tanner graph”, *Proc. Allerton Conf. on Commun., Control and Computing*, Urbana Champaign, IL, Sept. 2005.

[10] Y. Kou, H. Tang, S. Lin, and K. Abdel-Ghaffar, “On Circulant Low Density Parity Check Codes,” *Proc. IEEE Intl Symp. Inform. Theory,* p. 200, June 2002.

[11] J. W. Byers, M. Luby and M. Mitzenmacher, “A digital fountain approach to asynchronous reliable multicast,” *IEEE J. on Selected Area in Commun.*, 2002.