

Non-Binary Joint Network/Channel Coding for Multi-User ARQ

Ragnar Thobaben
ACCESS Linnaeus Center
Royal Institute of Technology (KTH)
SE-100 44 Stockholm, Sweden

Abstract—We propose a novel joint network and channel coding scheme for multi-user ARQ where network coding is employed to combine retransmissions to several users in a packet-based wireless system. Our approach features a joint design of non-binary network coding and trellis-coded modulation and enables symbol-wise iterative network/channel decoding. Our results based on Monte-Carlo simulations for Rayleigh block-fading channels show an improved performance compared to the conventional separate design while at the same time the amount of feedback and the complexity per user are reduced.

I. INTRODUCTION

Automatic repeat-request (ARQ) schemes play an important role in cellular wireless communication systems. Due to the time-variant nature of the transmission environment, the link adaptation is often based on imperfect channel state information; i.e., critical parameters may be outdated and/or available only in a coarse resolution. Transmission failures due to mismatched coding and modulation are the consequence; efficient techniques combining forward error correction coding with ARQ (so-called hybrid ARQ schemes) are desired.

Hybrid ARQ schemes which are for instance based on incremental redundancy provide a flexible solution to the above problem. If applied to a packet-based transmission scenario, their efficiency can however be reduced since entire frames are retransmitted where possibly only a fraction would be required. To overcome this drawback, a multi-user ARQ (MU-ARQ) scheme was proposed in [1]. Here, the idea is to combine retransmissions to several users in a single data packet by using network coding. This becomes possible if *all* users have decoded and acknowledged *all* previously transmitted packets. Based on the reported feedback the base station retransmits linear combinations of data packets (carried out on a finite field). This approach may hence be regarded as a network-coding solution to the MU-ARQ problem. Related work was recently presented in [2] where multi-layer ARQ strategies for the broadcast channel are discussed.

An extension of network-coded MU-ARQ to joint network/channel coding (JNCC) with iterative decoding was presented in our previous work [3]. Here, the network coded packet forms together with the pure data packet a linear block code, which is serially concatenated with the physical-layer channel code. By introducing interleavers and adjusting the components, a code structure is obtained which enables

iterative network/channel decoding (INCD). Significant performance gains compared to [1] were demonstrated while at the same time the complexity per user and the amount of feedback were reduced.

While the work in [1], [3] considers only binary coding, we generalize in this paper the JNCC scheme from [3] to network coding carried out on the Galois field $\text{GF}(2^m)$. Combined with trellis-coded modulation as inner channel code, a bandwidth-efficient approach to JNCC is obtained. We propose an extended iterative network/channel decoder which is adapted to this new setup and operates on the symbols of the $\text{GF}(2^m)$. For a scenario where mobile users acknowledge their success in decoding with 1-bit feedback, we propose a random network-code design which takes into account the average signal-to-noise ratio in order to determine the optimal packet degree. Again, we are able to demonstrate that the complexity per user and the amount of feedback are greatly reduced compared to the conventional separate design while we obtain a significant performance gain at the same time.

Non-binary network coding for wireless scenarios was as well considered, e.g., in [4]. Note furthermore that the code design proposed in this paper may be interpreted as adaptive fountain coding (see e.g. [5]) which constructs a local code for a subset of the data based on feedback by the users.

The remainder of the paper is organized as follows: Sections II and III introduce the underlying transmission system and outline the idea of network-coded MU-ARQ. Joint network/channel coding and decoding are addressed by Section IV, and the system optimization is discussed in Section V. Our simulation results for Rayleigh block-fading channels are presented in Section VI and compared to the conventional approach. Conclusions are drawn in Section VII.

II. SYSTEM MODEL AND ASSUMPTIONS

We consider a transmission system where a base station (BS) serves a group of M mobile users U_i with individual data packets. The packets during one epoch are denoted as $\{\mathbf{D}^{(1)}, \dots, \mathbf{D}^{(M)}\}$ where for ease of notation and without loss of generality, we assume that each user U_i is associated with one data packet $\mathbf{D}^{(i)}$, the so-called *intended* packet for user U_i . The transmitted packets are given by \mathbf{T}_n . They either represent channel-coded versions of the data packets $\{\mathbf{D}^{(1)}, \dots, \mathbf{D}^{(M)}\}$ (for instance by using low-density parity-check or Turbo codes) or contain only additional redundancy

broadcasted. Note that even users which are not addressed can exploit the packet to decode further unintended packets in order to gain additional *a priori* knowledge.

It is important to mention that the performance of MU-ARQ with separate channel and network coding depends on the tradeoff between the code rate of the employed channel code and the number of retransmissions: a conservatively chosen channel code (with low rate) will lead to few retransmissions with potentially higher packet degrees. However, the system's throughput will be limited by the rate of the channel code. On the other hand, if the code rate is chosen too high, opportunities to combine retransmissions become rare, and a high number of retransmissions with a low packet degree is required. It is therefore essential that the base station adjusts the code rate properly to the average SNR.

IV. JOINT NETWORK AND CHANNEL CODING FOR ITERATIVE DECODING

A. Motivation

A joint design of the channel code and the network code can be motivated in different ways. From a code-design point of view we can argue that, although the network code together with the channel code forms a serial concatenated code, only local constraints are created since both component codes are not separated by an interleaver. By introducing an interleaver, a strong code with large memory and improved distance properties (see e.g. [6]) can be obtained.

From a decoding point of view, the drawback of the conventional design is that the channel is converted with the help of the channel code into a block-erasure channel; i.e., packets for which the instantaneous SNR is below a certain threshold are discarded regardless of whether they could still contribute to the decoding. Accordingly, a decoding strategy which is capable of utilizing soft information from undecodable packages seems to be advantageous.

These two aspects are taken into account by the code construction which is proposed in the following. We introduce interleavers between the network and the channel encoders. The interleavers allow us to perform iterative network/channel decoding as explained in Section IV-C.

B. Encoding

The code structure which we obtain when separating the channel code by an interleaver from the network code is shown in Figure 2. It is similar to the one which was previously introduced in [3] for the binary case but differs in some details since we are in this paper interested in the case of non-binary network and channel coding.

In contrast to the binary case, encoding is completely carried out on code symbols which are defined on the $\text{GF}(2^m)$. We therefore suggest to realize the channel coding for the packets by trellis-coded modulation (TCM) which is capable of encoding m -ary input symbols. This choice will be further motivated in Section V. Similar to the conventional approach, the data packets $D_m^{(i)}$ are channel encoded and transmitted during the first phase. In order to create a joint network/channel code

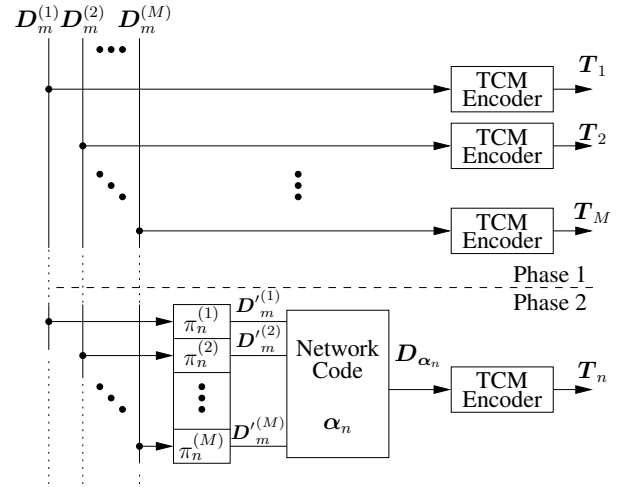


Fig. 2. Joint network and channel coding.

in the second phase, the $\text{GF}(2^m)$ representations $D_m^{(i)}$ of the data packets are permuted by *symbol* interleavers $\pi_n^{(i)}$ before the network coding with the encoding vector α_n is carried out. The interleaved data packets are then denoted by $D_m^{\prime(i)}$. Note that it is important that for each network-coded packet a new set of individual interleavers $\pi_n^{(i)}$ is generated. The overall code which is then obtained by this construction can be interpreted as a serially concatenated code with a specially structured interleaver. If only one network-coded packet is considered, the structure is similar to a serial concatenation of a non-binary single-parity-check (NB-SPC) code and the TCM. For the special case where we have $d_p = 1$ a Turbo-like code is constructed.

C. Iterative Network/Channel Decoding

Following the structure of the encoder, we can construct an iterative decoder which consists mainly of two different types of component decoders. The inner component decoders are given by soft-input/soft-output (SISO) TCM decoders, and the outer decoders correspond to the SISO decoders for the network code. Since the encoding vectors α_n of the network code are generated based on the previously received acknowledgements, the code structure is time variant, and the iterative network/channel decoder has to be adaptive. In the following, we will first explain the resulting decoder structure and detail later the iterative decoding.

1) *Decoder Structure*: In order to illustrate the construction of the iterative decoder we consider an example where $M = 5$ users are served by the base station. The transmissions to the users U_1 , U_3 , and U_4 fail during the first phase, and the base station generates the encoding vectors

$$\begin{aligned} \alpha_6 &= [\alpha_{61} , 0 , \alpha_{63} , \alpha_{64} , 0], \\ \alpha_7 &= [0 , 0 , \alpha_{73} , \alpha_{74} , 0], \\ \alpha_8 &= [0 , 0 , \alpha_{83} , 0 , 0], \end{aligned}$$

with $d_{p6} = 3$, $d_{p7} = 2$, and $d_{p8} = 1$, in order to combine retransmissions to the users (see as well [3]). This implies that the user U_1 is error-free after the first retransmission,

user U_4 after the second, and that user U_3 requires a third retransmission. The resulting decoder is depicted in Figure 3.

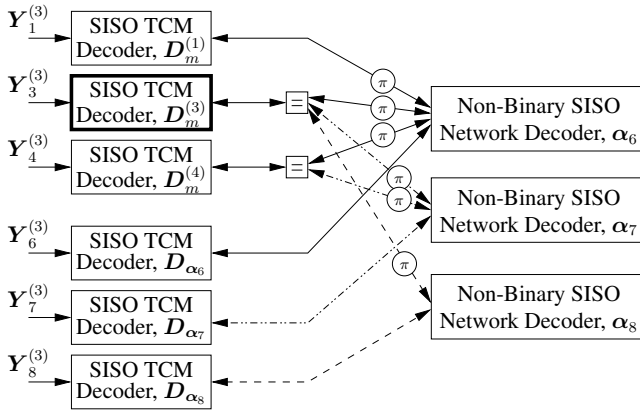


Fig. 3. Iterative network and channel decoder for user U_3 .

The decoding schedule is determined by the order of incoming network-coded packets. After receiving the first network-coded packet, the structure is similar to the serial concatenation of the TCM with a NB-SPC code. The SISO network-decoder corresponds accordingly to a NB-SPC decoder. With every additionally received network-coded packet, another SISO NB-SPC decoder is added to the decoder. It is connected to the TCM decoders as specified by the encoding vector α_n . For data packets which are included in several network-coded packets, the information which is provided by the network decoders has to be combined before it is passed to the TCM decoder. In Figure 3 this is illustrated by the variable nodes which connect the TCM decoders with the network decoders.

2) *Component Decoders and Exchange of Extrinsic Information:* While the structure of the iterative network/channel decoder is not significantly changed compared to the binary case, the realization of the component decoders and the exchange of extrinsic information become more involved. While in the binary case vectors of log-likelihood ratios are exchanged, symbol-wise decoding requires the exchange of matrices of log-probabilities.

Symbol-wise SISO TCM decoding is commonly realized by applying the BCJR algorithm [7] to the underlying trellis. Assuming that an *a priori* matrix A_n is available to the TCM decoder which is associated with the n -th transmitted packet T_n , the decoder provides us then with a matrix L_n of logarithmized *a posteriori* probabilities (Log-APPs) with elements $\{L_n\}_{ij} = \log(\Pr(D_{nj} = i | Y_n, A_n))$; i.e., it contains the Log-APPs for all realizations i of the symbols D_{nj} in the input vector D_n of the TCM encoder. The matrix of extrinsic information E_n is obtained by subtracting the *a priori* matrix A_n from the Log-APP matrix L_n , and we have $E_n = L_n - A_n$. The extrinsic information E_n is now handed over to the SISO NB-SPC decoder considering the encoding vector α_m where it becomes the *a priori* information $A_n^{(\alpha_m)}$ after permuting the columns as specified by the de-interleaver.

Exploiting the interpretation of the network code as a NB-SPC code, we can conveniently realize the SISO NB-SPC decoder by the BCJR algorithm. To explain this, let us (for ease

of notation) assume that the $GF(2^m)$ symbols $D^{(1)}, \dots, D^{(d_p)}$ are combined with the coefficients $\alpha_1, \dots, \alpha_{d_p}$ and produce the parity-check symbol $D^{(d_p+1)}$ as explained in equation (2); i.e., the code word of the NB-SPC code is given by $[D^{(1)}, \dots, D^{(d_p)}, D^{(d_p+1)}]$. We can now obtain a trellis representation by defining the trellis state S_l at code-symbol position l as the sum of the first l summands of the linear combination (2), and we get a state recursion

$$S_l = \sum_{i=1}^l \alpha_i D^{(i)} = S_{l-1} + \alpha_l D^{(l)}, \quad \text{with } S_0 = 0.$$

Note that with $\alpha_{d_p+1} = 1$ we get a terminated trellis with the final state $S_{d_p+1} = 0$. Applying the BCJR algorithm to this trellis representation is straightforward; it provides us again with a matrix of Log-APPs from which we can compute the extrinsic information by subtracting matrix of *a priori* information which the decoding is based on. In order to perform SISO network decoding of the entire packets, the SISO NB-SPC decoder is applied element-wise across the network-coded packets. The *a priori* matrices provided by the SISO TCM decoders have to be rearranged accordingly.

The processing at the variable nodes is finally very similar to the binary case: the extrinsic information at the output of one port is obtained by the sum of the *a priori* matrices which are available at the inputs of the remaining ports.

V. SYSTEM OPTIMIZATION

As explained above, the joint network-channel code is constructed adaptively by the base station based on the feedback from the users. Since the encoding of already transmitted packets cannot be changed, the adaption of the entire code can only be controlled by the network code. The goal of our code design is therefore to choose the inner channel code such that it offers the base station maximum flexibility. We suggest a TCM encoder which is used for all packets regardless of the average SNR. The network code can then be adapted to the inner channel code taking into account the average quality of the transmission channels.

The optimization of the network and the channel code is in the following based on an analysis of the convergence behavior of the iterative network/channel decoder with the extrinsic information transfer characteristics (EXIT charts, see e.g. [8]). Since the iterative decoder from Section IV-C is in general difficult to analyze, we aim in our optimization at achieving an error-free transmission after the first retransmission. This strategy can be justified since it mitigates high delays due to retransmissions.

A. Trellis-Coded Modulation

The efficiency of hybrid ARQ schemes is best reflected by the throughput which is obtained by the system. However, due to the serial concatenation of channel coding and network coding, the throughput is limited by the rate R_{tcm} of the TCM. Another upper bound on the throughput is given by the channel capacity C constrained on the signal-space constellation.

$$A = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad C = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}, \quad \text{and} \quad D = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (3)$$

In order to be able to approach the upper bound given by the constrained capacity for a given signal-space constellation, we propose to use a TCM where the size of the input alphabet equals the size of the output alphabet, $M_{in} = M_{out}$. This is beneficial especially for high SNR where conventional designs with $M_{in} < M_{out}$ will directly lead to a rate loss by a factor M_{in}/M_{out} . Furthermore it is known from the design of serially concatenated codes that the inner code has to be recursive to exploit the so-called interleaver gain [6] and to enable convergence of the iterative decoder. A TCM which fulfills these requirements and shows a desirable behavior over a wide SNR range is given in (3). It is described in its binary representation as a memory-4 convolutional encoder with 4 (binary) inputs and outputs (which is equivalent to an alphabet size of $M_{in} = M_{out} = 16$ symbols) by the matrices of its state-space representation. We use this encoder together with a 16-QAM constellation with Ungerböck labeling.

Remark: The choice of the TCM encoder has important consequences for the MU-ARQ protocol, the feedback strategy, and the decoding complexity. Since the proposed design leads to a situation where we have $R_{tcm} \geq C$, packets which are transmitted during the first phase carry not enough redundancy to be sufficiently protected. Depending on the individual channel conditions for the users, most packets will hence not be decodable. The users have therefore only to buffer the unintended packets for further use during the second phase instead of decoding them and acknowledge only successful decoding of the intended packets. Compared to [1] the number of packets which have to be decoded during the first phase is reduced by the number of users. The decoding complexity is further reduced since only a trellis code has to be decoded instead of an LDPC or Turbo code.

B. Network Code

For the approach from above the base station has now to choose a suitable set of encoding vectors α_n . Based on the acknowledgements, the base station creates a list of users which have to be considered by the retransmissions. Since in our system only 1-bit feedback for the intended packets is available as explained in the previous section, the base station lists the users in a random order and selects the users whose retransmissions are combined from the top of the list.

The problem of generating the encoding vector α_n consists now of (1) finding the packet degree d_p for the network-coded packets for which convergence of the iterative network-channel decoder is achievable and (2) generating suitable coefficients α_{ni} for the linear combination.

From an analysis of the EXIT charts of the network decoder for encoding vectors with fixed packet degree d_p but with varying coefficients α_{ni} we can conclude that the behavior of NB-SPC codes does not significantly change with different realizations of the encoding vector. Accordingly, assuming that

the packet degree d_p is given, picking the non-zero coefficients $\alpha_{ni} \in \text{GF}(2^m) \setminus 0$, with $i \in \mathcal{A}_n$, of the encoding vector at random will lead to an appropriate network-code design.

Considering now only a single retransmission, we can give the code rate of the network code as $R_{NC} = d_p/(d_p + 1)$. We can use this relation to give an upper bound on the packet degree for achieving an error free transmission similar to [3],

$$d_p \leq d_{max}, \quad \text{with} \quad d_{max} := \lfloor \frac{C}{R_{tcm} - C} \rfloor, \quad (4)$$

which holds for $C/R_{tcm} \geq 1/2$. For $C/R_{tcm} < 1/2$, more than one retransmissions are required (i.e., we have $d_p = 1$), and the network-coded MU-ARQ scheme collapses to a single-user ARQ (SU-ARQ) scheme with incremental redundancy, which forms a parallel concatenated code based on the TCM encoders.

In order to obtain finally the exact packet degree d_p the network code has to be matched to the EXIT charts of the TCM from (3) as demonstrated in Figure 4 for NB-SPC codes over the $\text{GF}(2^m)$, with $m = 4$. Here, the communication channel is realized by a Rayleigh fast-fading channel with $E_s/N_0 \in \{10, 15, 20\}$ dB. Note that the choice of the channel model is only a rough approximation for the actual scenario which however leads to good results.

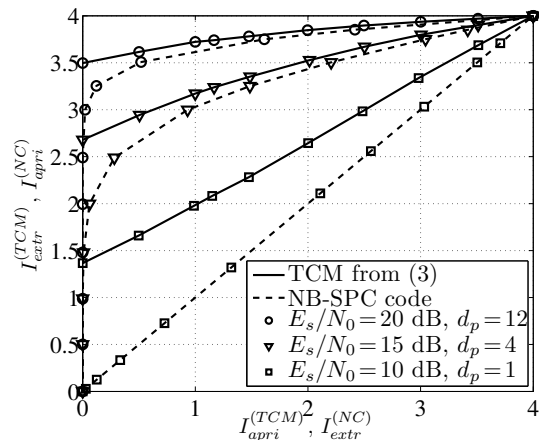


Fig. 4. EXIT charts for non-binary single-parity-check codes with degree d_p (dashed lines) and the TCM encoder from (3) for fast fading Rayleigh channels with $E_s/N_0 \in \{10, 15, 20\}$ dB (solid lines).

C. Protocol for Joint Network/Channel Coded MU-ARQ

We summarize the results from this section in a protocol enabling joint network-channel coded MU-ARQ: during the first phase, the users decode and acknowledge only their intended packets; the noisy observations of unintended packets are stored to enable further processing during the second phase. Based on the acknowledgements, the base station schedules the users which have to be addressed by retransmissions, and it determines the packet degree d_p based on the knowledge of the

average SNR. During the second phase, the base station generates a degree- d_p encoding vector α with random coefficients as explained above and combines the retransmission to the first d_p users on the list. The addressed users perform iterative network/channel decoding and send back acknowledgements for the intended packets. The base station continues with combined retransmissions until all users are error free.

VI. NUMERICAL RESULTS

The performance in terms of throughput (bits per channel use, bpcu) for the proposed joint network-channel code design with iterative decoding (JNCC+ID) is evaluated in this section based on simulations for the Rayleigh block-fading channel. The results are compared to an implementation of the conventional multi-user ARQ scheme (MU-ARQ) from [1] with separate network and channel coding and decoding and to the upper bound given by the ergodic capacity constrained on the used signal-space constellation (16-QAM in this case). Additionally, we consider a single-user ARQ scheme (SU-ARQ) which constructs a multiple Turbo code with each retransmitted packet. To allow a fair comparison, channel coding for conventional MU-ARQ is realized by an iteratively decoded serial concatenation of non-binary repetition or SPC codes (depending on the target rate) and the TCM proposed in Section V-A. A packet length of $K = 400$ bits is considered, and the channel coefficients are fixed for a block length of $L_f = 100$ channel uses. The cases of $M \in \{10, 20, 40\}$ users are addressed. The results are presented in Figure 5.

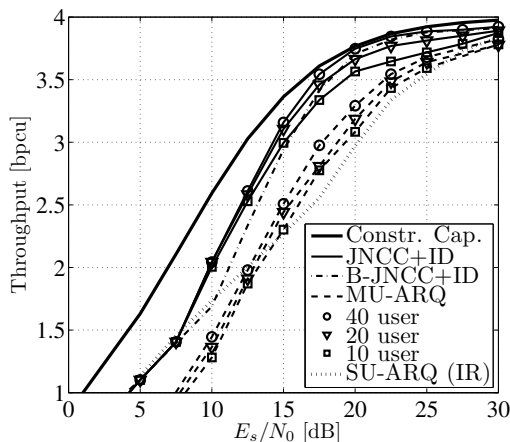


Fig. 5. Average throughput for the Rayleigh block-fading channel with $L_f = 100$ and $M \in \{10, 20, 40\}$.

As Figure 5 shows the proposed JNCC+ID scheme approaches the upper bound given by the constrained capacity in the SNR range $E_s/N_0 \in [17.5, 25]$ dB for a number of $M = 40$ users while for $E_s/N_0 \geq 25$ dB the achieved rate of the system saturates. In general, we can observe that as soon as the transmission of combined packets becomes possible (for $E_s/N_0 \geq 10$ in this experiment) the throughput is significantly increased compared to the single-user case (SU-ARQ). With an increasing number of users the system becomes more effective. The reason is that for a large number of users the

base station is able to transmit a higher relative number of packets with maximum packet degree. If a lower number of users is served, fewer opportunities occur to combine users at the maximum packet degree. This can be seen for instance in the 10-user case for $E_s/N_0 \geq 20$ dB where the maximum packet degree exceeds the number of users.

Compared to the conventional MU-ARQ with separate channel and network coding, the proposed JNCC+ID scheme leads to a significant improvement in performance. It benefits from the fact that the joint design and the iterative decoder allows all users to exploit the information from previously transmitted packets while in the conventional scheme the users have to discard packets which they were not able to decode.

Finally, in order to demonstrate the benefits of non-binary network-coding over binary network coding, Figure 5 shows as well the result for $M = 40$ users for a joint network-channel coding scheme where network coding is performed by “XORing” the data packets and iterative decoding is performed bit-based rather than symbol-based as proposed in this paper (B-JNCC+ID). The gain in E_s/N_0 exceeds 1 dB.

VII. CONCLUSION

We have proposed a new joint network and channel coding (JNCC) scheme for multi-user ARQ (MU-ARQ) in packet-based cellular networks. It features as a novelty non-binary network coding combined with trellis-coded modulation (TCM) and symbol-wise iterative network/channel decoding. We have demonstrated a system design for the special case where the users are restricted to 1-bit feedback in order to communicate their decoding state. For this setup we propose a random network-coding approach which is adapted to the average signal-to-noise ratio of the users while the same TCM encoder is used for all scenarios. Our results have shown that the proposed JNCC approach outperforms the conventional MU-ARQ significantly. At the same time, the amount of feedback and the decoding complexity are greatly reduced.

REFERENCES

- [1] P. Larsson and N. Johansson, “Multi-user ARQ,” in *Proc. IEEE Vehicular Technology Conference, 2006 (VTC 2006-Spring)*, May 2006, vol. 4, pp. 2052–2057.
- [2] A. Steiner and S. Shamai, “Multi-layer broadcasting hybrid-ARQ strategies for block fading channels,” *IEEE Trans. on Wireless Communications*, vol. 7, no. 7, pp. 2640–2650, July 2008.
- [3] R. Thobaben, “Joint network/channel coding for multi-user hybrid-ARQ,” in *Int. ITG Conference on Source and Channel Coding (SCC)*, Ulm, Germany, Jan. 2008.
- [4] M. Xiao and M. Skoglund, “M-user cooperative wireless communications based on nonbinary network codes,” in *Proc. IEEE Information Theory Workshop on Networking and Information Theory, 2009*.
- [5] M. Luby, “LT codes,” in *Proc. IEEE Symp. Foundations of Computer Science (FOCS)*, Vancouver, BC, Canada, Nov. 2002, pp. 271–280.
- [6] S. Benedetto, D. Divsalar, G. Montorsi, and F. Pollara, “Serial concatenation of interleaved codes: Performance analysis, design, and iterative decoding,” *IEEE Trans. Inf. Theory*, vol. 44, no. 3, pp. 909–926, May 1998.
- [7] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, “Optimal decoding of linear codes for minimizing symbol error rate,” *IEEE Trans. Inf. Theory*, pp. 294–287, Mar. 1974.
- [8] S. ten Brink, “Convergence behavior of iteratively decoded parallel concatenated codes,” *IEEE Trans. Commun.*, vol. 49, no. 10, pp. 1727–1737, Oct. 2001.