Nonlinear Trellis Codes for Binary-Input Binary-Output Multiple Access Channels With Single-User Decoding

Miguel Griot *Student Member, IEEE*, Andres I. Vila Casado *Student Member, IEEE*, Wen-Yen Weng *Student Member, IEEE*, Herwin Chan *Student Member, IEEE* and Richard D. Wesel *Senior Member, IEEE*.

Abstract

This paper presents a practical solution that provides uncoordinated access to a family of Binary-Input Binary-Output Multiple-Access Channels (BIBO-MACs), including the OR channel. We propose a solution using Interleaver-Division Multiple Access (IDMA) with single-user decoding. This solution features Nonlinear Trellis Codes (NLTC), that permit low-complexity decoding in support of high-speed applications. We present a design technique for nonlinear trellis codes with controlled ones densities for binary asymmetric channels (BACs), and in particular the Z-Channel, that arise in the BIBO-MAC when single-user decoding is used. Union bound techniques that predict the performance of these codes are also presented. Information theoretical calculations of the achievable sum-rate of these channels are presented as well. Simulation results and from a working FPGA implementation verify the performance and feasibility of the proposed algorithm.

Keywords

Channel coding, information rates, multiuser channels.

I. Introduction

THERE have been many approaches to providing multiple users access to the same channel. However, many common forms of multiple access, such as time-division (TDMA), frequency-division (FDMA), code-division (CDMA) or rate-splitting [1], require considerable coordination. A joint trellis-code design for all users has been proposed in [2], but this also requires coordination as one distinct channel code is assigned to each user. On the other hand, most popular forms of uncoordinated multiple access channels, such as Aloha, slotted Aloha, CSMA, and CSMA-CD, do not provide a clear QoS in terms of delay or delay jitter. One recent successful approach for uncoordinated multiple-access that does not introduce delay jitter is Interleaver-Division Multiple-Access (IDMA) [3], which uses interleaving to distinguish among signals from different users.

This work was supported by the Defence Advanced Research Project Agency under SPAWAR Systems Center San Diego Grant N66001-02-1-8938.
M. Griot, A. I. Vila Casado, H. Chan, and R. D. Wesel are with the Electrical Engineering Department, University of California, Los Angeles, CA 90095 USA (e-mail:[mgriot,avila,herwin,wesel]@ee.ucla.edu).
Wen-Yen Weng is with Ralink Technology Corporation, Hsin-Chu, Taiwan (wenyen@ralinktech.com.tw).

This work explores the applicability of the IDMA approach to a family of Multiple Access Channels with Binary Inputs and Binary Output (BIBO-MAC). In particular, we consider the OR Multiple Access Channel (OR-MAC), or its isomorphic channel, the Binary Multiplier Channel [4], as a target application for IDMA. Completely uncoordinated transmissions using IDMA and simple decoding that treats all signals except the desired signal as noise can theoretically achieve about 70% of the sum capacity over the OR channel for any number of users. By sacrificing 30% of the sum rate, this IDMA approach provides a significant reduction in complexity over coordinated transmission or joint decoding approaches, making it a practically attractive technique. We also explore a more general subset of the BIBO-MAC, which we call the OR Channel with Cancelations (ORC-MAC).

For the OR-MAC and ORC-MAC, IDMA requires channel codes with very low ones densities. To allow decoding at high speeds, this paper investigates Nonlinear Trellis Codes (NLTC) [2] with a controlled ones density. These codes, while having low latency and simple decoding, achieve a reasonable efficiency for any number of users. Turbo solutions, which more closely approach the sum-capacity, at the cost of more latency and complexity in the decoding, have also been explored in [5].

Section II reviews uncoordinated multiple access in the BIBO-MAC, and in particular the OR-MAC and ORC-MAC. Section III presents an NLTC design technique for this application. Section IV shows an analysis of the performance of these codes for large number of users, presenting an analytical tool to choose the proper number of states for the trellis code. Section V introduces a transfer-function bound for NLTCs operating on the Z-Channel and more generally on any binary asymmetric channel (BAC). Section VI presents performance results, and Section VII concludes the paper.

II. The BIBO-MAC Model

The BIBO-MAC can be modeled as follows: there are N users $\{U_1, \cdots, U_N\}$ transmitting N binary symbols (bits) $\{x_1, \cdots, x_N\}$. Denote the received binary symbol as y. Denote by ψ_m the conditional probability of receiving a 0 given that the sum of the transmitted bits is

m:

$$\psi_m = P\Big(y = 0\Big|\sum_{i=1}^{i=N} x_i = m\Big), \quad m \in \{0, \cdots, N\}. \tag{1}$$

The set of parameters $\{\psi_m : m = 0, ..., N\}$ depend on the particular BIBO-MAC.

A. Uncoordinated access to the BIBO-MAC

The interest of this work is to apply IDMA on the BIBO-MAC in a completely uncoordinated manner. Thus, timesharing is not allowed. For simplicity, we assume all the users are transmitting all the time at the same rate and all the users transmit with the same ones density $P(x_i = 1) = p$, for all $i = 1, \cdots, N$. Given a certain ones density p, the achievable sum of the rates of all the N users, which we will call sum-rate, is:

$$R_{+N}(p) \leq H\Big(\sum_{m=0}^{m=N}\binom{N}{m}p^m(1-p)^{N-m}\psi_m\Big) - \sum_{m=0}^{m=N}\binom{N}{m}p^m(1-p)^{N-m}H(\psi_m), \tag{2}$$

where $H(\cdot)$ is the entropy function. To maximize the sum-rate, the optimal ones density is then:

$$p^{opt} = \text{argmax}_{p\in[0,1]}\{R_{+N}(p)\}. \tag{3}$$

Depending on the particular values ψ_m, the sum-rate can take values from 0 (for the case $\psi_m = 1/2, \forall m$) to 1 (upper bound since the output is binary). This approach may not be capacity achieving, since greater rates could be achieved with timesharing. However, we will investigate cases where timesharing is not necessary to achieve capacity or where the loss in rate due to not using timesharing is small.

B. BIBO-MAC with single-user decoding

The BIBO-MAC sum-rate capacity may be achieved with joint decoding of all the transmitted sequences. However, joint decoding is very complex, especially for a large number of users. In high-speed applications where joint decoding is unavailable for complexity reasons, Single-User Decoding (SUD) must be used. With single-user decoding, each user treats all but the desired signal as noise, transforming the BIBO-MAC into the Binary Asymmetric

Channel (BAC) with cross-over probabilities:

$$\begin{aligned} P(y=1|x=0) = \alpha = 1 - \sum_{i=0}^{N-1} \tbinom{N-1}{i} p^i (1-p)^{N-1-i} \psi_i, \\ P(y=0|x=1) = \beta = \sum_{i=0}^{N-1} \tbinom{N-1}{i} p^i (1-p)^{N-1-i} \psi_{i+1}. \end{aligned} \tag{4}$$

The achievable sum-rate for N users under single-user decoding is:

$$R_{+N}^{SUD} = N \cdot \Big[H\big\{(1-p)(1-\alpha) + p\beta\big\} - (1-p)H(\alpha) - pH(\beta) \Big]. \tag{5}$$

C. The OR-MAC

In the OR-MAC, if all users transmit a 0, then the channel output is a 0. However, if one or more users transmit a 1, then the channel output is a 1. The OR channel can be used as a simple communications model that describes the multiple-user local area network optical channel with non-coherent combining. For short distances, the effect of noise can be considered negligible. A 1 is transmitted as light and a 0 is transmitted as no light. If any user transmits light, light is received. Only when all users do not transmit light, a 0 is received. In this simple model it is assumed that there is no destructive interference between users.

The OR-MAC is a particular case of the BIBO-MAC channel presented above where:

$$\text{OR-MAC} : \psi_m = \begin{cases} 1 & \text{for} \quad m = 0, \\ 0 & \text{for} \quad m = 1, \cdots, N. \end{cases} \tag{6}$$

It can be derived from Eq. (2) and Eq. (3) that the capacity region for N users is given by the constraint $R_{+N} = R_{+\infty} \leq 1$ even without time-sharing [1], using the optimal ones density:

$$\text{OR-MAC} : p^{opt}(N) = 1 - (1/2)^{1/N}. \tag{7}$$

Note that the achievable sum-rate is 1 regardless of the number of users N. This means that if joint decoding is employed, completely uncoordinated transmission on the OR-MAC is theoretically possible with the same efficiency as TDMA, for any number of users.

When single-user decoding is used over the OR-MAC, each user perceives a particular case of the BAC, commonly known as the Z-Channel, where:

$$\alpha = 1 - (1-p)^{N-1}, \quad \beta = 0. \tag{8}$$

The achievable sum-rate becomes:

$$R_{+N}^{SUD}(p) = N \cdot \Big[H\big\{(1-p)^N\big\} - (1-p)^{N-1} H\big\{(1-p)^{N-1}\big\}\Big]. \tag{9}$$

An interesting property of the OR-MAC is that with single-user decoding its maximum theoretical sum-rate R_{+N}^{SUD} monotonically decreases only to $\ln 2 \simeq 0.6931$ as the number of users N increases. This is a relatively small loss in rate for the substantial reduction in complexity. Also, the optimal ones density is practically the same as the ones density for joint decoding. Namely:

$$p^{opt}(N) \to \big(1 - (1/2)^{1/N}\big), \text{ for SUD as } N \to \infty. \tag{10}$$

A sketch of the proof of the asymptotic sum rate and the asymptotically optimal ones density for SUD on the OR-MAC is as follows: [1] First, we prove that R_{+N}^{SUD} is monotonically decreasing. Let $p(N+1)$ be the optimal ones density for $N+1$ equal-rate users. Compute the possibly suboptimal ones density $\tilde{p}(N)$ for N equal-rate users as that which satisfies $(1-\tilde{p}(N))^N = (1-p(N+1))^{N+1}$. Using Eq. 9, then $R_{+N}^{SUD}\big(\tilde{p}(N)\big) > R_{+(N+1)}^{SUD}\big(p(N+1)\big)$. Thus the sum rate is monotonically decreasing because a possibly sub-optimal ones density for N users leads to a higher symmetric sum rate than the optimal ones density for $N+1$ users. To prove that the limit is $\ln 2$, prove that for a fixed number of users N, $R_{+N}^{SUD}(p)$ is quasi-concave as a function of p, and since $R_{+N}^{SUD}(0) = R_{+N}^{SUD}(1) = 0$, there exists one (and only one) local maximum which is the global maximum. Then prove that the conjectured $p^{opt}(N)$ of Eq. (10) tends to the local maximum when $N \to \infty$. Finally, plug the right-hand expression of Eq. (10) on Eq. (9), and make $N \to \infty$ to find that the limit is $\ln 2$.

Fig. 1 shows the maximum theoretical sum-rate using both joint decoding, which is always

[1]Note to reviewers: full proof of these statements can be provided upon request. Due to the limitation in space, we only show here the sketch of the proofs.

1, and single-user decoding, which decreases to $\ln 2$. It also shows the maximum theoretical sum-rate of single-user decoding using a ones density of $p = 0.5$, which rapidly decreases to zero as the number of users increases. The poor performance of the $p = 0.5$ ones density demonstrates that codes with low ones densities are required for this application.

D. The ORC-MAC

We now extend the OR-MAC to handle the possibility of destructive interference, as might occur in optical multiple access. As with the OR channel, when all users transmit a 0, a 0 is received. Also, when only one user transmits a 1, a 1 is received. However, when more than one user transmits a 1, there is a certain probability (associated with a destructive interference event) that a 0 is received. This probability is always less than 1/2 and depends on the number of ones transmitted. We call this subset the OR-MAC with cancelations, or ORC Multiple Access Channel (ORC-MAC).

Following the notation introduced in Section II-A, the ORC-MAC can be expressed as a general BIBO-MAC with the following constraints:

$$\text{ORC-MAC} : \begin{cases} \psi_0 = 1, \psi_1 = 0, \\ 1/2 \geq \psi_m \geq \psi_{m'}, \quad \forall m' \geq m \geq 2 \end{cases} \tag{11}$$

As a specific example of an ORC-MAC consider the Coherent Interference MAC (CI-MAC), for which:

$$\text{CI-MAC:} \begin{cases} \psi_0 = 1, \psi_1 = 0, \\ \psi_m = P\Big(\Big| \sum_{i=1}^{i=m} e^{j\theta_i} \Big|^2 < \sigma\Big), \quad \forall m \geq 2, \end{cases} \tag{12}$$

where $\theta_i \sim U[0, 2\pi)$ are random variables with uniform distribution, and σ is a threshold which which will be considered 1/2 in this work.

The maximum achievable sum rate of the ORC-MAC, with time-sharing, is 1. To see this, note that if the outputs of all but one user are set to 0, then the maximum achievable rate for that user is 1. Hence, with time sharing, any combination of rates with sum-rate equal to 1 can be achieved. For the ORC-MAC, it can be proven that the sum-rate capacity, for both joint and single-user decoding, is lower bounded by a strictly positive number regardless of the number users, as stated in the following theorems.

Theorem II.1: Using a ones density of the form

$$p(N) = 1 - \delta^{1/N}, \tag{13}$$

the achievable sum rate on the ORC-MAC is lower bounded by:

$$R_{+N} \geq \max_{\delta \in [1/2,1]} \Big\{ H\Big(\delta + \psi_2 h(\delta)\Big) - \psi_2 h(\delta) \Big\}, \tag{14}$$

for any number of users N, where $h(\delta) = 1 - \delta(1 - \ln \delta)$.

Theorem II.2: Using a ones density of the form shown in Eq. (13), the achievable sum rate on the ORC-MAC with single-user decoding is lower bounded by:

$$R_{+N}^{SUD} \geq \begin{array}{c} \max_{\delta \in [1/2,1)} \Big\{ \delta \cdot \log(1/\delta) \cdot (\psi_2 \cdot \log(1/\delta) - 1) \cdot \log_2 \left(\frac{1-g(\delta)}{g(\delta)} \right) \\ + \log(1/\delta) \cdot H\left(g(\delta)\right) - \log(1/\delta) \cdot H\left(\psi_2 \cdot (1-\delta)\right) \} \end{array}, \tag{15}$$

for any number of users N, where $g(\delta) = \delta - \psi_2[1 - \delta(1 - \ln \delta)]$.

The proofs are sketched as follows: First prove that Eq. (2) and Eq. (5) are decreasing with N for fixed ψ_i's. Given a certain value of ψ_2, consider the worst case scenario $\psi_m = \psi_2$, $\forall m \geq 2$. Then use Eq. (13) in Eq. (2), and let $N \to \infty$ to prove Theorem II.1. Use Eq. (13) in Eq. (4) and Eq. (5), and make $N \to \infty$ to prove Theorem II.2.

We have corroborated empirically that the ones density of the form expressed in Eq. (13), using the value of δ that maximizes Eq. (14), is indistinguishably close to the optimal ones density. The same is true for single-user decoding. Fig. 2 shows the lower bounds of the sum-rate capacity, for both joint decoding and single-user decoding, depending on ψ_2. It also shows the actual sum-rate for the 200-user case for the case where $\psi_m = \psi_2$ for all $m \geq 2$ (this is the worst possible channel given ψ_2). The lower bound is actually very tight in that case. Note that the smaller ψ_2 the less sum-rate is lost compared to 1 (the maximum achievable sum-rate using time-sharing). Also, as ψ_2 goes to 0, and hence the ORC-MAC tends to the OR-MAC, the sum-rate lower bounds tend to OR-MAC achievable sum-rates, both for joint decoding and single-user decoding. Fig. 3 shows the optimal δ's for the ones densities for both joint decoding and single-user decoding. Note that when ψ_2 goes to 0, δ tends to 1/2 for both joint decoding and single-user decoding, resulting in the optimal ones density for the OR-MAC.

III. NLTC with Controlled Ones Density

Papers appearing since the 1950's have addressed the problem of designing codes with $p = 0.5$ for the binary asymmetric channel, and in particular the Z-Channel. See [6] for a unified account on such codes and [7] for the most recent advances in this field. Only recently there has been work on LDPC codes with an arbitrary density of ones, see [8] and [9].

In this section we present a design technique for trellis codes with an arbitrary ones density for the binary asymmetric channel and specifically for the Z-Channel. We use a conventional rate-$1/n_0$ feed-forward encoder in order to determine the branches of the trellis, but instead of using generator polynomials to compute the output of each branch, a nonlinear lookup-table directly assigns the output values to each branch [2].

The parameters of the trellis codes are chosen as follows: Denote the desired target sum-rate as R_+, and the optimal ones density as p. Then, the number of outputs per trellis section n_0, as a function of the number of users N, is:

$$n_0(N) = N/R_+, \tag{16}$$

where we assume that R_+ is chosen so that n_0 is a natural number.

In order to provide the required ones density, each of the branches needs to have the proper Hamming weights (W_h). Using a 2^ν-state encoder ($B = 2^{\nu+1}$ branches), and an optimal ones density of p, there should be B_w branches with Hamming weight $w = \lfloor p \cdot n_0 \rfloor$ and $B_{w+1} = B - B_w$ branches with Hamming weight $w + 1$, where B_w should be chosen to minimize the deviation (Δ) from the desired ones density:

$$\Delta = |p \cdot n_0 - (B_{w+1} \cdot (w + 1) + B_w \cdot w)/B|. \tag{17}$$

Given the Hamming weights of each branch, the remaining task is to assign the positions of the ones in each label. We provide a design technique that assigns these positions with the goal of maximizing the minimum distance between codewords. This design technique uses a non-standard definition of distance specific to the Z-Channel and the BAC. We introduce this distance in the next section.

A. Directional Hamming Distance and ML decoding for the Z-Channel

Consider any two words of length n bits, $X = \{x_1, \cdots, x_n\}$ and $\tilde{X} = \{\tilde{x}_1, \cdots, \tilde{x}_n\}$. Define the Directional Hamming Distance $d_D(X, \tilde{X})$ as the number of positions where $x_i = 0$ and $\tilde{x}_i = 1$, with $i = 1, \cdots, n$. Note that $d_D(X, \tilde{X})$ is not necessarily equal to $d_D(\tilde{X}, X)$. Denote the received word as $Y = \{y_1, \cdots, y_n\}$. It is clear that given Y, any possible transmitted codeword X on the Z-Channel must satisfy $d_D(Y, X) = 0$, since there cannot be any one-to-zero transitions. The most likely transmitted codeword $\hat{X}$, is the codeword that minimizes $d_D(X, Y)$ the number of zero-to-one transitions, among those codewords X satisfying $d_D(Y, X) = 0$. Hence, the ML decoder for the Z-Channel chooses the codeword $\hat{X}$ as:

$$\hat{X} = \text{argmin}_{X \in \mathcal{N}} \Big[d_D(X, Y) \Big], \tag{18}$$

where $\mathcal{N}$ is the set of codewords that satisfy $d_D(Y, X) = 0$.

Let α be the probability of a zero-to-one transition in the Z-Channel. Using Eq. (18), it can be derived that the probability of transmitting X and decoding $\tilde{X}$ under ML decoding is:

$$P_e(X \to \tilde{X}) = \begin{cases} \frac{1}{2} \cdot \alpha^{d_D(X, \tilde{X})} & , W_H(X) = W_H(\tilde{X}) \\ \alpha^{d_D(X, \tilde{X})} & , W_H(X) < W_H(\tilde{X}) \\ 0 & , W_H(X) > W_H(\tilde{X}). \end{cases} \tag{19}$$

where $W_H(\cdot)$ denotes the Hamming weight. If two codewords have different Hamming weights, the codeword with the smaller Hamming weight will never be incorrectly decoded by a maximum likelihood (ML) decoder when the code with the larger Hamming weight is transmitted. On the other hand, if both codewords have the same Hamming weight, the directional Hamming distances are equal and errors can be made in either direction. In any case, the directional distance that matters is the larger of the two. Thus, a proper definition of pair-wise distance for the Z-Channel is:

$$d_Z(X,\tilde{X}) = \max[d_D(X,\tilde{X}), d_D(\tilde{X},X)] \tag{20}$$

This metric for the Z-Channel is well known, appearing in [6] and [7] among other papers.

B. Pessimistic definition of distance over the Z-Channel

The definition of distance for the Z-Channel cannot be applied branch-wise, since it is impossible to tell from an individual branch which codeword will end up having more Hamming weight. For that reason, we will use a pessimistic definition of distance for our trellis code design, considering both directional distances. Namely, the safest definition of branch-wise distance between any two branches b_i and b_j would be

$$d_p = \min[d_D(b_i,b_j), d_D(b_j,b_i)]\,, \tag{21}$$

which is the pessimistic branch-wise metric that will be maximized in our design.

With this branch-wise metric, codewords with equal Hamming weights produce larger values of d_p than codewords with different Hamming weights, so we will assign output values to the trellis branches with as similar Hamming weight as possible, preferably equal.

C. Pessimistic definition of distance over the BAC

For the case of the BAC, with zero-to-one transition probability α and one-to-zero transition probability β, the ML decoder chooses the codeword $\hat{X}$ as:

$$\hat{X} = \text{argmin}_X \Big[\ln\Big(\frac{1-\beta}{\alpha}\Big) d_D(X,Y) + \ln\Big(\frac{1-\alpha}{\beta}\Big) d_D(Y,X) \Big]. \tag{22}$$

Consider two codewords of length n, X and $\tilde{X}$. The pair-wise error probability can be expressed as:

$$P_e(X \to \tilde{X}) = h(\alpha,\beta,X,\tilde{X}) \sum_{r=0}^{d_D(X,\tilde{X})} \sum_{s=0}^{d_D(\tilde{X},X)} \Bigg[$$

$$I\Bigg((2r - d_D(\tilde{X},X)) \ln\left(\frac{1-\beta}{\alpha}\right) + (2s - d_D(X,\tilde{X})) \ln\left(\frac{1-\alpha}{\beta}\right) > 0 \Bigg) \left(\frac{\alpha}{1-\alpha}\right)^r \left(\frac{\beta}{1-\beta}\right)^s \Bigg], \tag{23}$$

where $I(\cdot)$ is the indicator function and

$$h(\alpha,\beta,X,\tilde{X}) =$$
$$(1-\alpha)^{n-W_H(X)}(1-\beta)^{W_H(X)} \left[\frac{\left(\alpha/(1-\alpha)\right)^{n-W_H(X)-d_D(X,\tilde{X})} - 1}{1-(1-\alpha)/\alpha} \right] \left[\frac{\left(\beta/(1-\beta)\right)^{W_H(X)-d_D(\tilde{X},X)} - 1}{1-(1-\beta)/\beta} \right]. \tag{24}$$

There is no trivial dependence between the directional distances $d_D(X,\tilde{X})$ and $d_D(\tilde{X},X)$, and $P_e(X \to \tilde{X})$. However, this work considers channels with great asymmetry, where β is much smaller than α. In that case, the situation may be approximated by the previous discussion. Thus, when designing the NLTC, the same pessimistic definition of distance is used for both the Z-Channel and the BAC.

D. Nonlinear trellis code design

As mentioned before, the code design consists of assigning output values to the branches of the trellis. Those outputs have to maintain the desired average ones density p. The trellis paths of two valid codewords split from a common state at some trellis section, and merge to a common state at some other trellis section. Since a feed-forward encoder is used, two valid codewords must traverse different branches produced by a common input in ν consecutive trellis sections before a merge. The design procedure begins by ensuring all branches produced by the same input to have a pessimistic distance (d_p) of at least 1 between each other. Thus, if each of those sections adds at least 1 to the pessimistic distance, then $d_{\min} \geq \nu$. This can be accomplished if $\binom{n_0}{w} \geq 2^\nu$, where $w = \lfloor p \cdot n_0 \rfloor$. This last inequality is satisfied in the applications considered in this work, since the code-rates are very small (n_0 is large).

Once the weights of the branches are chosen, and a set of branch labels is selected to ensure

$d_p \geq 1$, we now assign branch labels to the branches. Our approach for this assignment is based on Ungerboeck's idea of maximizing the distance between splits and merges [10]. Given the Hamming weights in our design, the best we can hope for is to lower bound the pessimistic distances d_p between branches that share a split or merge by w (some may have distance $w+1$ between each other). Ungerboeck's rule can be extended more deeply into the trellis, and maximize not only the distance between splits, and the distance between merges, but the distance between the 4 branches emanating from a split in the previous trellis section, or the 8 branches emanating from a split two sections before, and so on. The same can be done with the merges moving backwards in the trellis. Notice that if for all sets of 8 branches emanating from a split two sections before their distance is lower bounded by w, the distance between all 4 branches emanating from any split a trellis section before, and every pair of branches at the beginning of any split are also lower bounded by w. The same idea applies to the merges. If we consider h sections after a split, and g sections before a merge, the new bound for the minimum pessimistic distance is

$$d_{\min} \geq (w-1) \cdot (h+g) + \nu + 1. \tag{25}$$

The sum $h+g$ is limited by the parameters of the design. First, $h+g \leq \nu+1$. All the branches have distance of at least w between each other when equality holds. Also, the sum $h+g$ is limited by constraints resulting from the requirement that h and g must be small enough that the relevant sets of branches in each of the h or g trellis sections of the split or merge are all separated by the maximum pessimistic distance. Note that the condition need only be enforced for the trellis section involving the most branches (i.e. the last section of a split or the earliest section of a merge). The condition will then automatically be satisfied by the trellis sections involving fewer branches since these smaller groups are themselves strict subsets of larger groups that meet the enforced condition.

From the splitting point of view, the largest groups contain 2^h branches, which should have pessimistic distance of at least w between each other. Satisfying $w \cdot 2^h \leq n_0$ is required to guarantee a pessimistic distance of w between any two branches in a group of 2^h branches.

From the merging point of view, the largest groups contain 2^g branches, and therefore the requirement is $w \cdot 2^g \leq n_0$. The last constraint is given by the fact that each branch belongs to one group of 2^h and one group of 2^g, and no pair of branches belongs to the same two groups.

As an example, consider a rate-1/8 ($n_0 = 8$) 8-state trellis ($\nu = 3$), where a ones density of $p = 1/4$ is required. Then, the Hamming weight of each output must be $w = n_0 \cdot p = 2$. There are $\binom{8}{2} = 28$ possible outputs with $w = 2$ and there are $2^{\nu+1} = 16$ branches. Hence, we can choose 16 different outputs with $d_p \geq 1$ between each other. Since $w = 2$, the maximum pessimistic distance between two outputs is 2. The maximum number of outputs with $d_p = 2$ between each other is $n_0/w = 4$. Therefore we can choose $h = g = 2$ so that $2^h = 2^g = 4$. For example, Fig. **??** shows the 4 branches produced by a split from the all-zero state in the previous trellis section. It also shows the 4 branches that can produce a merge to the all-zero state in the next trellis section. In general, $h = 2$ implies that we need to maximize the distance between every group of branches departing from the states (abX), with ab fixed for each group and $X = \{0, 1\}$, with any input (there are 4 in each group). To satisfy $g = 2$ we maximize the distance between any group of branches departing from states (XXc) with the same input (there are also 4 in each group). Table XXX shows a possible labeling that achieves that. The table is constructed so that outputs in a same row or a same column need to have $d_p = 2$. Therefore, using (25) the minimum distance of the code is $d_{\min} = 8$, which is the maximum possible minimum distance to achieve given the parameters of the example.

IV. Providing the same sum rate and performance for any number of users

With the nonlinear trellis codes of this paper, for a target sum-rate R_+, and a specified target BER, there may be a limitation on the number of users N if the number of states 2^ν is not large enough. This statement can be understood quantitatively as follows:
As seen in Section II-A, the optimal ones density for a certain number of users is very well approximated by Eq. (13), where δ depends on the channel (see Fig. 2). Then, the total number of ones in all the $2^{\nu+1}$ branches increases monotonically with the number of users

N, converging to a limit as follows:

$$W_b(N) \simeq 2^{v+1} n_0 p(N) \simeq \frac{(N(1-\delta^{1/N}))2^{v+1}}{R_+} \rightarrow \frac{\ln(1/\delta)2^{v+1}}{R_+}. \quad (26)$$

On the other hand, from Eq. (16) the number of output bits per trellis section linearly increases with N. Hence, for a large enough number of users, $n_0(N)$ becomes greater than $W_b(N)$. Let N_c denote the smallest number of users at which $n_0(N_c) \geq W_b(N)$. The design of the code for N_c is straightforward. For each branch, add ones in positions that aren't used in previous branches until its assigned Hamming weight is reached. Moreover, the best code for N_c users is essentially the best code for any number of users greater than N_c. The only difference is that as N grows more zeros are added to the output.

The channel degrades as N increases, hence degrading the code performance. However, for N_c sufficiently large, this degradation becomes marginal, as both α and β converge to fixed values. For example, in the OR-MAC, plugging Eq. (7) into Eq. (8), then:

$$\alpha(N) \simeq 1 - (1/2)^{(N-1)/N}, \quad (27)$$

which converges to 1/2. N_c increases with ν, so choosing ν sufficiently large (e.g. large enough to handle the $\alpha = 1/2$ case on the OR-MAC), a target sum rate can be achieved regardless of the number of users with essentially the same performance.

Fig. 5 shows the number of output bits per trellis section n_0 and the total number of ones in all the branches W_b vs. the number of users, for $\nu = 5$ and $\nu = 6$ codes designed for the OR-MAC, using a target sum rate of $R_+ = 0.3$ and $\delta = 1/2$. With $\nu = 5$ $N_c = 44$, and $\alpha(N_c) = 0.492$. With $\nu = 6$, $N_c = 89$, and $\alpha(N_c) = 0.496$, which is already very close to $\alpha(\infty) = 0.5$. The question is whether a code designed for $N_c = 44$ can continue to perform well as the number of users increases beyond 44 (with proper added zeros) and α increases beyond 0.492 towards 1/2. If not, can the $\nu = 6$ code designed for $N_c = 89$ continue to perform well as the number of users increases beyond 89 (with the proper added zeros to the output) and α increases beyond 0.492 towards 1/2?. As will be corroborated in Section VI, $\nu = 6$ is sufficient to achieve the target sum-rate of 0.3 with consistent performance regardless of the number of users.

V. Transfer Function Bound for NLTC Codes

Ellingsen [11] provided a combinatorial expression for an upper bound on the BER of linear block codes over the Z-channel under ML decoding. For convolutional codes assuming binary PAM or QPSK, Viterbi [12] introduced an analytical technique using generating functions to provide a union bound on the BER of convolutional codes. Viterbi's technique is based on a 2^{ν}-state diagram for the convolutional encoder. In the case of general trellis codes where high level constellations introduce nonlinearity, Biglieri [13][14] generalized Viterbi's algorithm by using the product state diagram with $2^{2\nu}$-states. Biglieri's algorithm can be applied to nonlinear trellis codes over the Z-channel, and more generally over the BAC, with modifications on the pairwise error probability measure.

A. Transfer Function Bound over the Z-Channel

The pairwise error probability of decoding X into $\hat{X}$ under ML decoding over the Z-Channel is shown in Eq. (19). Hence, the sum of the error probabilities of transmitting one sequence and decoding the other (or vice versa) is

$$P_e(X \to \hat{X}) + P_e(\hat{X} \to X) = \alpha^{\max(d_D(X,\hat{X}),d_D(\hat{X},X))} \leq \frac{1}{2}[\alpha^{d_D(X,\hat{X})} + \alpha^{d_D(\hat{X},X)}] \tag{28}$$

Therefore, if $P_e(X \to \hat{X})$ is replaced (not always upper-bounded) by $\frac{1}{2}\alpha^{d_D(X,\hat{X})}$ for all the codewords X and $\hat{X}$, the transfer function bound technique can be readily applied to the NLTC to yield a valid overall upper bound because of the additive property of the directional distance.

As in [13], the product state diagram consists of state pairs, (s_e, s_r), where s_e is the encoder state and s_r the receiver state. Following Biglieri's notation, the product states can be divided into two sets, the good states denoted by S_G and the bad states denoted by S_B defined as

$$S_G = \{(s_e, s_r) \mid s_e = s_r\},\ S_B = \{(s_e, s_r) \mid s_e \neq s_r\}. \tag{29}$$

By suitably renumbering the product states, we get the transition matrix

$$S(W,I) = \left[\begin{array}{c|c} S_{GG}(W,I) & S_{GB}(W,I) \\ \hline S_{BG}(W,I) & S_{BB}(W,I) \end{array}\right], \tag{30}$$

where the $N \times N$ matrix $S_{GG}(W,I)$ accounts for transitions between two good product states, the $N \times (N^2 - N)$ matrix $S_{GB}(W,I)$ accounts for transitions from good product states to bad product states, and so forth. N is the number of encoder states 2^{ν}.

For each transition in the product state diagram, $S_1 \to S_2$, the branch is labeled by

$$p(S_1 \to S_2) W^{d_D(x_e,x_r)} I^{d_H(u_e,u_r)}, \tag{31}$$

where $d_H(\cdot,\cdot)$ denotes the Hamming distance, u_e and x_e denote the input and output word for the encoder states respectively, and u_r and x_r denote the input and output word for the receiver states. The transfer function $T(W,I)$ is

$$T(W,I) = p_s\{S_{GG} + S_{GB}(I - S_{BB})^{-1} S_{BG}\}\mathbf{1}, \tag{32}$$

where $p_s = [\frac{1}{N}\frac{1}{N}\cdots\frac{1}{N}]$ is the marginal probability distribution of the encoder states and $\mathbf{1} = [\mathbf{11}\cdots\mathbf{1}]^{\mathbf{T}}$. The BER bound is computed as

$$BER \leq \frac{1}{2} \cdot \frac{1}{k} \cdot \left.\frac{\partial T(W,I)}{\partial I}\right|_{W=\alpha, I=1}. \tag{33}$$

B. Transfer Function Bound over the BAC

For the BAC, using a variation of the Bhattacharyya bounding technique [15], the sum of the error probabilities of transmitting either sequence and decoding the other can be upper bounded by:

$$\begin{aligned} &P\left(X \to \hat{X}\right) + P\left(\hat{X} \to X\right) = \textstyle\sum_Y \min\left\{P\left(Y|X\right), P\left(Y|\hat{X}\right)\right\} \\ &\leq \sum_Y \sqrt{P\left(Y|\hat{X}\right) P\left(Y|X\right)} = \prod_i \sum_{y_i} \sqrt{P\left(y_i|\hat{x}_i\right) P\left(y_i|x_i\right)}. \end{aligned} \tag{34}$$

Now,

$$\sum_{y_i} \sqrt{P(y_i|\hat{x}_i)P(y_i|x_i)} = \begin{cases} 1, & \text{if} \quad x_i = \hat{x}_i, \\ \sqrt{\alpha(1-\beta)} + \sqrt{\beta(1-\alpha)} & \text{if} \quad x_i \neq \hat{x}_i. \end{cases} \tag{35}$$

Therefore,

$$P\left(X \to \hat{X}\right) + P\left(\hat{X} \to X\right) \leq \left(\sqrt{\alpha(1-\beta)} + \sqrt{\beta(1-\alpha)}\right)^{d_H\left(X,\hat{X}\right)}. \tag{36}$$

Replacing the branch label of Eq. (31) by

$$p(S_1 \to S_2) W^{d_H(x_e,x_r)} I^{d_H(u_e,u_r)}, \tag{37}$$

the BER is upper bounded by

$$BER \leq \frac{1}{2} \cdot \frac{1}{k} \cdot \left. \frac{\partial T(W,I)}{\partial I} \right|_{W=\sqrt{\alpha(1-\beta)}+\sqrt{\beta(1-\alpha)},I=1}. \tag{38}$$

VI. Performance Results

We have tested the NLTC performance over the uncoordinated OR-MAC and CI-MAC with single-user decoding, for different numbers of users varying from 6 to 1500.

A. NLTC on the OR-MAC

Fig. 6 shows the BER of various 64-state NLTC codes designed to work in a 6-user OR-MAC, along with their theoretical transfer function bounds. The simulation performances of the following codes are shown: a rate-1/17 NLTC code with $p = 2/17$, a rate-1/18 NLTC code with $p = 1/8$ and a rate-1/20 NLTC code with $p = 1/8$. The ones densities of $p = 1/8$ and $p = 2/17$ are close to 0.108, the optimal density for a 6-users OR-MAC with single-user decoding.

The transfer function bounds are tight in all three cases. The transfer function bound is an upper bound on the expectation assuming an infinite decoder depth, and it is not unexpected for a simulation to be slightly above the bound (as in the case of the rate-1/17 code), since the decoding depth is 35 rather than infinite and there is variation of a simulation around the expectation.

Also, in order to prove that NLTC codes are feasible today for high speeds, a hardware demonstration was built using fiber optics and Xilinx Virtex2-Pro 2V20 FPGAs. The implementation had an equivalent gate count of 360K gates and is able to encode and decode the rate-1/20 NLTC code concatenated with a Reed-Solomon block code at an information rate of 70Mbps. A detailed description of the FPGA implementation can be found in [16].

Results for rate-1/20 NLTC code (not including the Reed-Solomon code) obtained in the FPGA testbed are also shown in Fig.6. Due to design constraints, the hardware Viterbi decoder has a maximum path distance metric of 20, and hence 1-to-0 transitions are given a distance of 20 instead of ∞. This difference causes the deviation from the theoretical bound at low bit error rates. Finally, Fig. 7 shows the BER of these codes in terms of the number of users present in an OR-MAC.

In Section IV, we introduced an analysis to decide the proper number of states of the trellis section. For the OR-MAC, a 64-state NLTC ($\nu = 6$) is enough to achieve similar performance at same sum-rate for any number of users. Table I shows BERs for 6 up to 1500 users. The performance is practically the same for all the cases.

B. Concatenation of NLTC code with a Block Code

A good solution for applications that require a very low BER is to include a high-rate block code that can correct a small number of symbol errors as an outer code, dramatically lowering the BER. As mentioned before, the OR channel can be used as a simple communications model that describes the multiple-user local area network optical channel with non-coherent combining. With the concatenation of NLTC codes with a Reed-Solomon Code a good part of the capacity is achieved, with a suitable BER for optics, and a feasible complexity for today's technology at optical speeds. We have built a 6-user optical system transmitting data on a single wavelength to demonstrate this last statement [16]. This implementation received the first prize at the 2006 student design contest sponsored by the ACM Design Automation Conference and the IEEE International Solid State Circuits Conference.

A concatenation of the rate-1/20 NLTC code with a (255-byte, 247-byte) Reed-Solomon code has been tested for the 6-user OR-MAC scenario. The rate of this code is $(247/255) \cdot$

$(1/20) \simeq 0.0484$, which gives a sum-rate of approximately 0.2906. The BER observed was 2.48×10^{-10}. Although simulations with the concatenated Reed-Solomon code for more than 6 users haven't been performed, it can be inferred from results on Section VI-A that the system proposed in this work can achieve almost 30% of full capacity, with a BER on the order of 10^{-10}, even for a large number of users.

C. NLTC for 6-user CI-MAC

Fig. 8 shows the performance of a 64-state NLTC code over the CI-MAC with single-user decoding. This code is a 1/30-rate NLTC (which gives a sum-rate of 0.2) with a ones density $p = 1/15$, and was designed for the 6-user CI-MAC, which with single-user decoding has a maximum achievable sum-rate of $R_{+6}^{SUD} \simeq 0.48$ with a ones density of $p \simeq 0.059$. Fig. 8 also shows the analytical bound. For the BAC, the analytical bound is not as tight as for the Z-Channel. The Bhattacharyya bounding technique of Eq. (36) used for the BAC is not as tight as Eq. (28), which was used for the Z channel. Table II shows the performance of 128-state NLTC codes for a sum-rate of 0.2, and different number of users.

VII. Conclusions

This paper addressed the problem of designing codes with low ones densities for the Z-channel and the binary asymmetric channel. These codes can be used with an IDMA-based architecture to allow uncoordinated multiple access in the BIBO-MAC, and in particular the OR-MAC and ORC-MAC. Achieving a low ones density (or any ones density below 0.5) requires the use of nonlinear codes. In this work, nonlinear trellis codes were designed to allow low decoder complexity so as to be computationally feasible today at high information rates.

The concatenation of these codes with high-rate block codes achieves a good part of the capacity of the channel with a low BER and a fast decoder. Design criteria for NLTC codes with controlled ones density were introduced.

Moreover, tight analytical bounds on their performance over the Z-Channel were presented. Also, an analytical bound using the Bhattacharyya bounding technique was shown for the BAC. Though not as tight as the bound for the Z-Channel, this bound still gives a good idea of the code's performance.

Also, an analysis on the performance of these codes for large number of users depending on the number of states of the trellis has been shown. This tool provides a means to decide the proper number of states of the trellis in order to achieve a similar performance for any number of users, with the same trellis structure. This is an interesting feature of this solution, and it makes it especially attractive for a large number of users, where coordination becomes an issue.

References

[1] A.J. Grant, B. Rimoldi, R.L. Urbanke, and P.A. Whiting. Rate-splitting multiple access for discrete memoryless channels. *IEEE Trans. on Info. Theory*, 47:873–890, Mar 2001.

[2] Pierre Chevillat. N-user trellis coding for a class of multiple-access channels. *IEEE Transactions on Information Theory*, January 1981.

[3] Li Ping, Lihai Liu, Keying Wu, and W.K. Leung. Interleave division multiple-access. *IEEE Transactions on Wireless Communications*, 5:938–947, April 2006.

[4] T.M. Cover and Joy A. Thomas. Elements of Information Theory. *Wiley Series in Telecommunications*, 1991.

[5] M. Griot, A.I. Vila Casado, and R.D. Wesel. Non-linear turbo codes for interleaver-division multiple access on the OR channel. In *GLOBECOM '06. IEEE Global Telecomm. Conf.*, 27 Nov. - 1 Dec. 2006.

[6] T. Kløve. Error correcting codes for the asymmetric channel. In *Dept. Mathematics, Univ. Bergen, Bergen, Norway, Tech. Rep.1809-0781*, 1995.

[7] Fang-Wei Fu, San Ling, and Chaoping Xing. New lower bounds and constructions for binary codes correcting asymmetric errors. *IEEE Trans. on Info. Theory*, 49:3294–3299, December 2003.

[8] A. Bennatan and D. Burshtein. On the application of LDPC codes to arbitrary discrete-memoryless channels. *IEEE Trans. on Info. Theory*, 50:417–438, March 2004.

[9] E.A. Ratzer and D.J.C. MacKay. Sparse low-density parity-check codes for channels with cross-talk. *Info. Theo. Workshop*, pages 127–130, April 2003.

[10] G. Ungerboeck. Channel coding with multilevel/phase signals. *IEEE Trans. on Info. Theory*, 28:55–67, January 1982.

[11] P. Ellingsen, S. Spinsante, and O. Ytrehus. Maximum Likelihood Decoding of Codes on the Asymmetric Z-channel. In *The 10th IMA International Conference on Cryptography and Coding 2005.*

[12] A. Viterbi. Convolutional codes and their performance in communication systems. *IEEE Trans. on Comm.*, 19:751–772, Oct 1971.

[13] E. Biglieri. High-level modulation and coding for nonlinear satellite channels. *IEEE Trans. on Comm.*, 32:616–626, May 1984.

[14] Y.-J. Liu, I. Oka, and E. Biglieri. Error probability for digital transmission over nonlinear channels with application to TCM. *IEEE Trans. Info. Theory*, 36:1101–1110, Sep 1990.

[15] Miguel Griot, Wen-Yen Weng, and Richard D. Wesel. A tighter bhattacharyya bound for decoding error probability. *IEEE Communications Letters*, 11, April 2007.

[16] H. Chan, M. Griot, A. Vila Casado, R. Wesel, and I. Verbauwhede. High speed channel coding architectures for the uncoordinated OR channel. In *IEEE 17th INTERNATIONAL CONFERENCE ON Application-specific Systems, Architectures and Processors (ASAP)*, Steamboat Springs, Colorado, 2006.

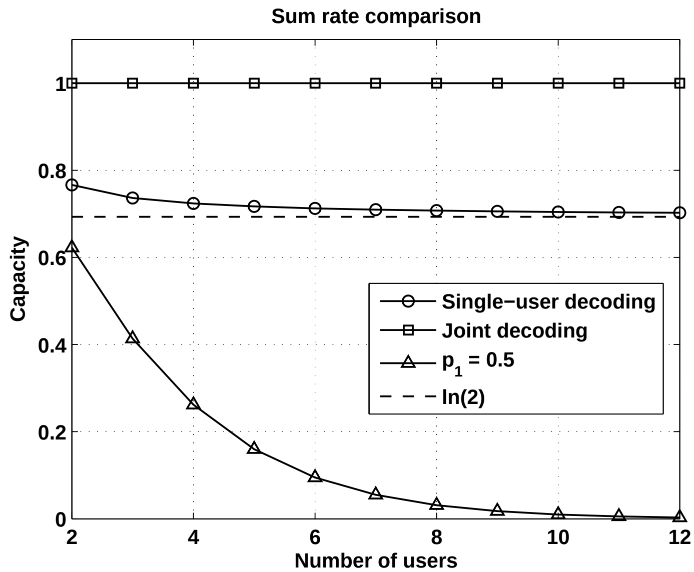

Fig. 1. Maximum OR-MAC theoretical sum-rates as a function of the number of users for joint decoding, single-user decoding and joint decoding with a ones density of 0.5.

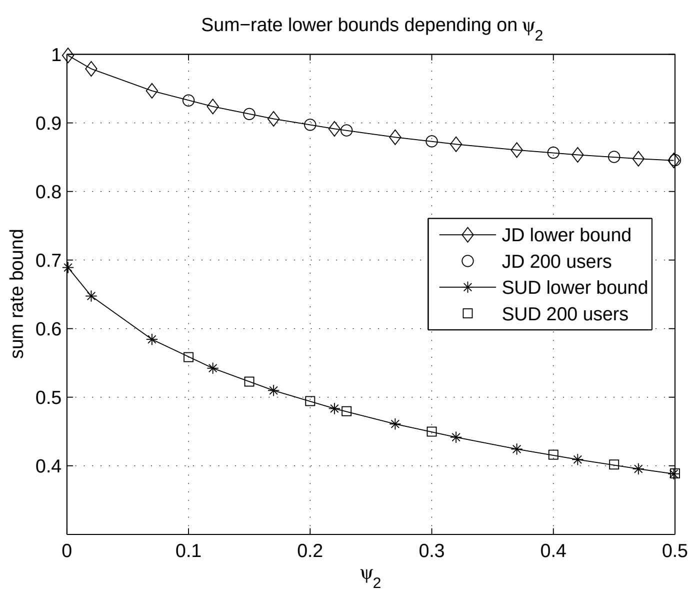

Fig. 2. Bit Error Rate (BER) lower bound for the ORC-MAC, with joint decoding (JD) and single-user decoding (SUD), depending on ψ_2

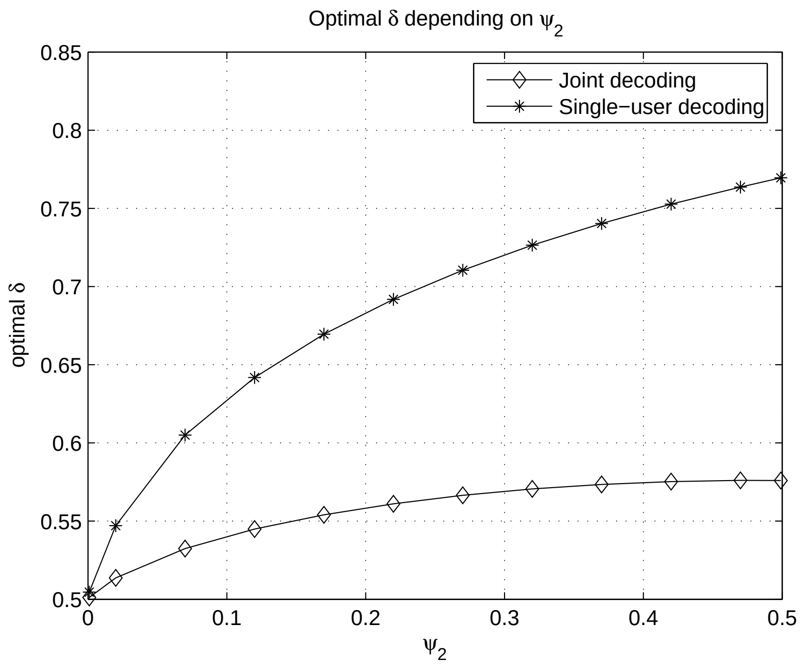

Fig. 3. Optimal δ for the ORC-MAC, with joint decoding (JD) and single-user decoding (SUD), depending on ψ_2

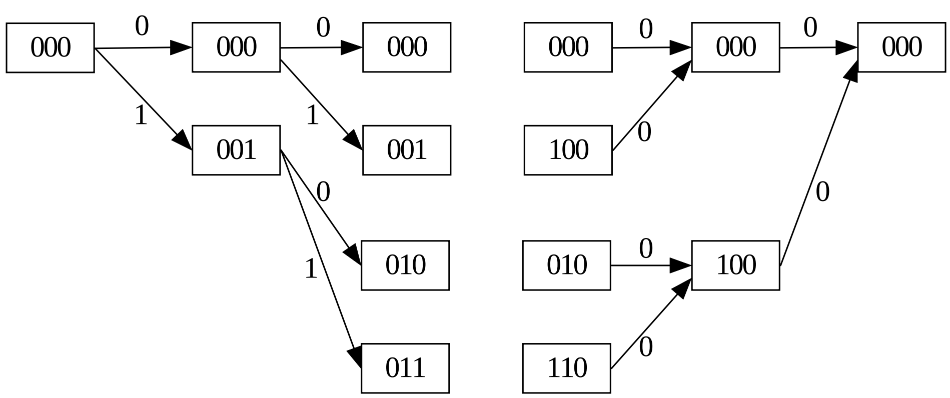

Fig. 4. Example of Ungerboeck's extension. The 4 branches produced by a split in the all-zero codeword the previous trellis section and the 4 branches going to a merge to the all-zero codeword in the next trellis section.

TABLE I

Example of labeling design. S denotes the current state, u denotes the input bit. Each output corresponds to the branch produced by the input u when the current state is S.

S	u	output	S	u	output	S	u	output	S	u	output
000	0	11000000	010	0	00101000	100	0	00010001	110	0	00000110
000	1	00110000	010	1	10000010	100	1	01000100	110	1	00001001
001	0	00001100	011	0	01000001	101	0	00100010	111	0	10010000
001	1	00000011	011	1	00010100	101	1	10001000	111	1	01100000

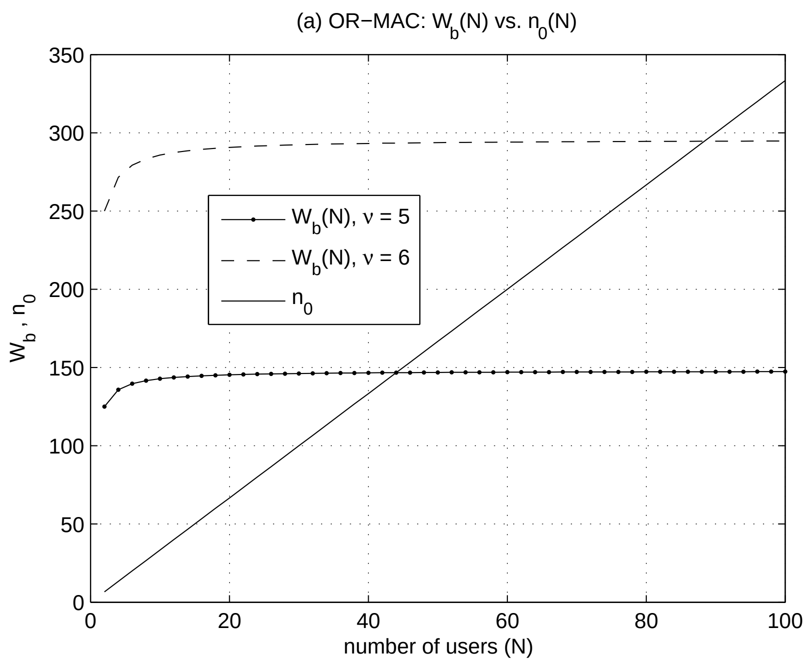

Fig. 5. Total number of ones in the output of all branches (W_b) and number of output bits per branch (n_0) vs. number of users (N), for the (a) OR-MAC, (b) CI-MAC, for different number of states 2^{ν}.

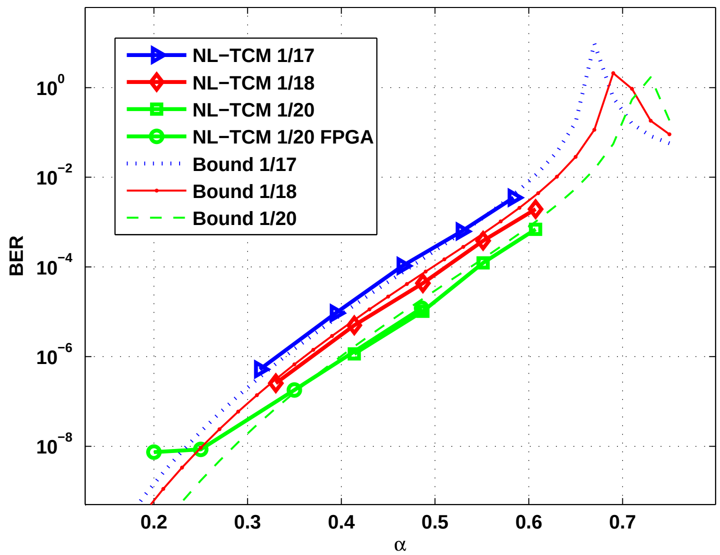

Fig. 6. Bit Error Rate (BER) of NLTC codes versus Z-Channel crossover probability (α)

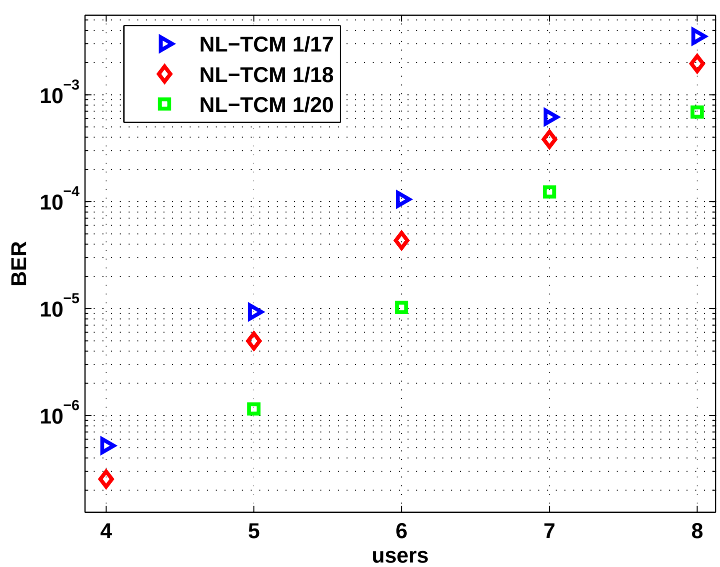

Fig. 7. BER of NLTC codes vs. number of users

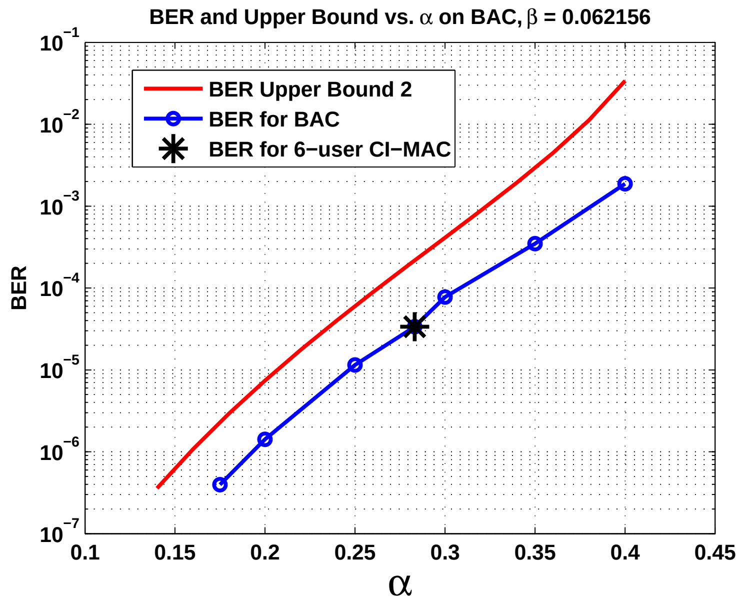

Fig. 8. BER of 64-state NLTC codes vs. α over the BAC

TABLE II

Performance of 64-state NLTC for different number of users (N) and a sum-rate $R_+ 0.3$, on the OR-MAC.

N	n_0	R_+	α	BER
6	20	0.3	0.439	1.0214×10^{-5}
100	344	0.291	0.4777	1.1046×10^{-5}
300	1000	0.3	0.4901	1.2157×10^{-5}
900	3000	0.3	0.4906	1.2403×10^{-5}
1500	5000	0.3	0.4907	1.2508×10^{-5}

TABLE III

PERFORMANCE OF 128-STATE NLTC FOR DIFFERENT NUMBER OF USERS (N) AND A SUM-RATE $R_+ = 0.2$, ON THE CI-MAC.

N	R_+	α	β	BER
6	0.2	0.2832	0.0622	1.46×10^{-5}
32	0.2	0.3107	0.0664	2.71×10^{-5}
104	0.2	0.3147	0.0677	6.35×10^{-5}