# Soft Information Relaying for Wireless Networks with Error-Prone Source-Relay Link

Petra Weitkemper[1], Dirk Wübben[1], Volker Kühn[2], Karl-Dirk Kammeyer[1]
[1]University of Bremen, Department of Communications Engineering, Otto-Hahn-Allee 1, 28359 Bremen, Germany
[2]University of Rostock, Institute of Communications Engineering, Richard-Wagner-Str. 31, 18119 Rostock, Germany
Email: {weitkemper, wuebben, kammeyer}@ant.uni-bremen.de and volker.kuehn@uni-rostock.de

## Abstract

Relays in wireless networks can be used to decrease transmit power while additionally increasing diversity. Distributed turbo coding as a special case of *decode-and-forward* is very powerful in relay networks when assuming error free decoding in the relay. In practical wireless networks, however, this assumption is only justifiable if an ARQ protocol is applied which leads to lower throughput. *Soft-reencoding* and transmission of the reliability of reencoded bits helps the destination to decode the message. Reencoding in the relay with a recursive convolutional code as used for turbo-codes, can lead to error propagation. In this paper the effects of error propagation in relay networks are investigated and more suitable distributed coding schemes are presented for soft-reencoding. As the often used assumption of Gaussian distributed disturbance at the destination is not valid for the considered system setup, the calculation of Log-Likelihood-Ratios (LLR) for the received noisy reliability information is derived analytically.

## 1 Introduction

An important advantage of relay stations in wireless networks is the potential to reduce the required overall transmit power due to exponentially decreasing path loss. Applying the well known strategy *decode-and-forward* (DF) instead of simple *amplify-and-forward* (AF), additional coding gain can be obtained [1]. A powerful DF-scheme is the distributed turbo-code (DTC) introduced in [2], where the source encodes the information bits with a systematic, recursive convolutional code $\mathcal{C}_s$ and broadcasts these coded bits to the destination and to the relay. The relay decodes the information by a decoder $\mathcal{D}_s$, interleaves the hard estimated information bits, reencodes them with recursive code $\mathcal{C}_r$ and transmits these code bits to the destination. Thus, the two component codes can be decoded like a common turbo-code at the destination and in case of error-free decoding at the relay, the performance will be similar to the standard turbo-code. However, in difference to a common turbo-code the component codes $\mathcal{C}_s$ and $\mathcal{C}_r$ are placed at distributed positions, so that the input bits for $\mathcal{C}_r$ have to be estimated. This leads to the challenging task to deal with erroneously or unreliably decoded bits at the relay. In order to cope this task *soft-output encoder* were proposed for the encoder in the relay that makes use of the soft decoder output of $\mathcal{D}_s$ to calculate soft-values for the *reencoded* bits [3], [4]. The transmitted reliability information can then be incorporated into the overall turbo-like decoder at the destination and consequently even erroneous codewords may improve the overall performance. As demonstrated in Section 3.1 the recursive structure of the code $\mathcal{C}_r$ applied in [3] suffers from the effect of error propagation. In this case the reliability of the recursively soft encoded bits depend strongly on the least reliable input bits so far in the whole block. Although the DTC is a consequent application of the well known turbo-codes, other distributed codes may lead to better results if the special network topology is taken into account. For that reason different distributed convolutional coding schemes with soft-reencoding are presented and compared in this paper. On the one hand, recursive structures in the relay are avoided as already done in [4] and on the other hand, the distribution of the constituent codes is changed over the network. Performance and robustness are as well considered as computational complexity.

The paper is organized as follows: The system model of the relay network is introduced in Section 2 and the basics of distributed coding are explained. Section 3 deals with relaying soft reencoded bits: The calculation of LLRs for convolutionally soft encoded bits will be explained as well as the calculation of LLRs of noisy soft-bits at the receiver. The scenarios and coding schemes are explained and compared via simulations in Section 4. Conclusions are given in Section 5.

## 2 System Model and Basics of Distributed Coding Schemes

We consider the two-hop relay system shown in Fig. 1. The source ($s$) encodes the information bit vector[1]

$$\mathbf{b} = (b_1, b_2, ..., b_{N_u}) \quad (1)$$

using the code $\mathcal{C}_s$ and transmits the resulting BPSK

[1]Throughout the paper vectors are denoted as bold letters and elements as italic letters, e.g. $\mathbf{b}$ and $b$. Soft estimates are identified with a tilde $\tilde{b}$ and hard estimates by $\hat{b}$.

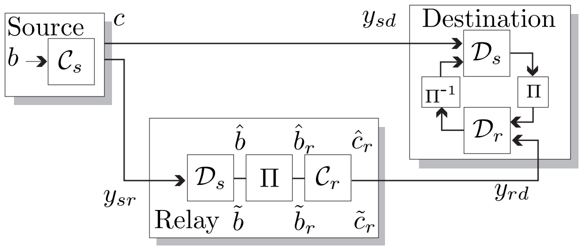


Fig. 1. Block diagram of a two-hop relay system

modulated code bit vector

$$\mathbf{c} = (c_1, c_2, ..., c_N) \ . \tag{2}$$

to the destination ($d$) and the relay ($r$). In order to ensure interference free transmission multiple access schemes like TDMA or FDMA are assumed for the source and the relay channel. Furthermore, a real-valued system model is considered for the derivations but the extension to the complex case, e.g. for QPSK, is straight forward. The application of higher constellations like 16-QAM for the soft-reencoded relay-destination link is an ongoing research topic. The received signal vectors at the destination and at the relay are denoted as $\mathbf{y}_{sd}$ and $\mathbf{y}_{sr}$, respectively. They are disturbed by independent frequency-flat fading channels characterized by the path loss $g_{sd}, g_{sr}$, the channel coefficients $h_{sd}$, $h_{sr}$ with mean power 1 and additional white Gaussian noise $\mathbf{n}_{sd}$ and $\mathbf{n}_{sr}$ of variance $\sigma_{sd}^2$ and $\sigma_{sr}^2$, respectively. For notational simplicity an overall gain of $G_{xy} = P_x \cdot g_{xy}$ is defined, where $P_x$ denotes the corresponding transmit power. The discrete time baseband received signals can then be written as

$$\begin{aligned} \mathbf{y}_{sr} &= \sqrt{G_{sr}} h_{sr} \cdot \mathbf{c} + \mathbf{n}_{sr} \\ \mathbf{y}_{sd} &= \sqrt{G_{sd}} h_{sd} \cdot \mathbf{c} + \mathbf{n}_{sd} \ . \end{aligned} \tag{3}$$

In a *distributed coding* (DC) scheme the received signal $\mathbf{y}_{sr}$ is hard-output decoded at the relay to obtain the estimated information bits $\hat{\mathbf{b}}$. The interleaved information bits $\hat{\mathbf{b}}_r$ are then reencoded by a second code $\mathcal{C}_r$ and the resulting code bits $\hat{\mathbf{c}}_r$ are transmitted to the destination yielding the receive signal

$$\mathbf{y}_{rd} = \sqrt{G_{rd}} h_{rd} \cdot \hat{\mathbf{c}}_r + \mathbf{n}_{rd} \ . \tag{4}$$

Thus, DC is a special case of decode-and-forward, as different codewords are transmitted by source and relay due the different codes or interleaver and consequently iterative decoding can generally be performed at the destination.

A special variant of the DC approach is the *distributed turbo-code* (DTC), where systematic and non-systematic recursive convolutional codes are used as constituent codes in the source and in the relay, respectively. The authors in [3] used for example $\mathcal{C}_s = [1,\ 5/7]_8$ and $\mathcal{C}_r = [5/7]_8$ to construct the overall DTC $[1,\ 5/7;\ 5/7]_8$. Except the fact that the encoding is distributed over the network this code is similar to a standard turbo-code. However, as will be demonstrated in Section 3, this scheme is sensitive to decoding errors in the relay. If decoding errors occur at the relay the codeword $\hat{\mathbf{c}}_r$ transmitted to the destination suggests a wrong information bit sequence. If additionally the relay-destination link is more reliable than the direct link, it is likely that the overall decoding at the destination decides in favor of the code word transmitted by the relay and consequently the decoding fails. In this case the additional information from the relay is destructive.

In order to overcome this problem, a promising approach called DTC-SIR (*DTC with soft information relaying*) was presented in [3]. The basic idea is to use the soft-output $\tilde{\mathbf{b}}$ of the decoder $\mathcal{D}_s$ in the relay to calculate soft estimates $\tilde{\mathbf{c}}_r$ for the code bits of code $\mathcal{C}_r$ by a *soft-input soft-output (SISO) encoder*. As explained in the next section this *soft-reencoding* can be achieved by a modified BCJR algorithm.

# 3 Soft information relaying

## 3.1 Calculation of soft code bits at the relay

The first step for soft information relaying is the calculation of the a-posteriori Log-Likelihood-Ratio (LLR) $L(b|\mathbf{y}_{sr})$ for information bit $b$ based on the received sequence $\mathbf{y}_{sr}$ using the common BCJR algorithm [6] with respect to code $\mathcal{C}_s$. Another equivalent representation of the reliability information is given by the expectation value of the corresponding bit

$$\tilde{b} = \mathrm{E}\{b|\mathbf{y}_{sr}\} = \tanh\left(L(b|\mathbf{y}_{sr})/2\right) \ , \tag{5}$$

called *soft bit* in the literature. Subsequently these bits are interleaved by $\Pi$ and the resulting sequence $\tilde{b}_r$ is fed to the soft-output encoder for $\mathcal{C}_r$. The challenging task of this reencoder is to perform the encoding in the relay on basis of $\tilde{b}_r$ in order to determine soft-values for the code bits $\tilde{c}_r$. It is worth to note that this reliability information can not be obtained directly from the decoder $\mathcal{D}_s$, as an interleaver and different codes $\mathcal{C}_s$ and $\mathcal{C}_r$ are applied to achieve a distributed coding scheme.

For a common convolutional encoder the input bits stem from GF(2), the output is given by an XOR operation of the register elements and the input bits depending on the generator polynomials. In contrast the SISO encoder has to compute the probabilities of $c_r =$ +1 and $c_r =$ -1 to determine the desired reliability for the code bits

$$\begin{aligned} \tilde{c}_r &= \tanh\left(L(c_r|\tilde{\mathbf{b}}_r)/2\right) \\ &= P(c_r = +1|\tilde{\mathbf{b}}_r) - P(c_r = -1|\tilde{\mathbf{b}}_r) \ . \end{aligned} \tag{6}$$

To calculate for example the probability $c_r = +1$ all possible input vectors $\mathrm{GF}(2)^{N_u}$ have to be considered that would lead to the code bit +1. The basics of this soft-reencoding for convolutional codes were presented in [3] and will be given in this section based on a trellis representation of the convolutional code. In addition, the implementation as a modified BCJR algorithm is suggested here.

**a) SISO Reencoding by modified BCJR**

Considering a trellis representation of the convolutional code, the values in the shift registers represent the state of the trellis and the input bits specify the branch taken from each state. The probability of an output bit to be +1 can be calculated by

$$P(c_r = +1|\tilde{\mathbf{b}}_r) = \sum_{\substack{(S',S) \\ c(S',S)=+1}} P(b(S',S)|\tilde{\mathbf{b}}_r) \cdot P(S'|\tilde{\mathbf{b}}_r) \,, \quad (7)$$

where $S'$ and $S$ represent the current and next trellis state, respectively. $\tilde{\mathbf{b}}_r$ denotes the interleaved soft estimates of the information bits. The information and code bit corresponding to the state transition $(S', S)$ from state $S'$ to state $S$ are denoted as $b(S', S)$ and $c(S', S)$, respectively. Given the current state $S'$, the probability of a certain transition is given by the probability of the corresponding information bit. The calculation for $P(c_r = \text{-}1|\tilde{\mathbf{b}})$ is equivalent to (7). The probability of the next state $S$ can be calculated by the recursive formula

$$P(S|\tilde{\mathbf{b}}_r) = \sum_{S'} P(b(S',S)|\tilde{\mathbf{b}}_r) \cdot P(S'|\tilde{\mathbf{b}}_r) \,. \quad (8)$$

Obviously, the calculations in (7) and (8) correspond to the forward recursion of a standard BCJR algorithm, where only the A-Priori-Probabilities given by $\tilde{b}_r$ but no channel observations are considered for the state transition probabilities. Accordingly, the soft-reencoding can be implemented as efficient as the BCJR.

Finally, the question arises, in which format the reliability information (6) should be transmitted by the relay. Instead of transmitting log-likelihood ratios (LLR) [5] we choose to transmit the soft code bits $\tilde{c}_r$ as the amplitude of the transmit signal is by definition in the range $\{\pm 1\}$ and the power constraint at the relay can be easier fulfilled. Consequently, the signal received by the destination from the relay is now given by

$$\mathbf{y}_{rd} = \sqrt{G_{rd}} h_{rd} \cdot \tilde{\mathbf{c}}_r + \mathbf{n}_{rd} \,. \quad (9)$$

**b) Behavior of soft reencoder for different codes**

Interestingly, the distribution of the soft reencoder output LLRs

$$L(c_r|\tilde{\mathbf{b}}_r) = \ln\left(\frac{P(c_r = +1|\tilde{\mathbf{b}}_r)}{P(c_r = \text{-}1|\tilde{\mathbf{b}}_r)}\right) \quad (10)$$

highly depends on the used channel code structure. In Fig. 2 a) the distribution over time and b) the histogram of the conditioned LLRs $L(c_r|\tilde{\mathbf{b}}) \cdot c_r$ for the recursive code $\mathcal{C}_r = [5/7]_8$ and $\text{SNR}_{sr} = 0$ dB are depicted. The reliability is dominated by the least reliable bit so far in the frame and is therefore decreasing over time in the long term. Furthermore, contrary to the usual assumption at the receiver, the distribution of the LLR is not Gaussian anymore. Further simulations

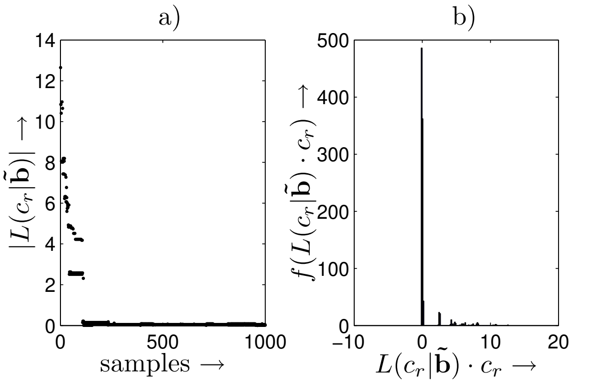


Fig. 2. Distribution of output LLRs over time (a) and histogram of conditioned LLRs (b) for $\mathcal{C}_r = [5/7]_8$, $\text{SNR}_{sr} = 0\,\text{dB}$, 1000 samples

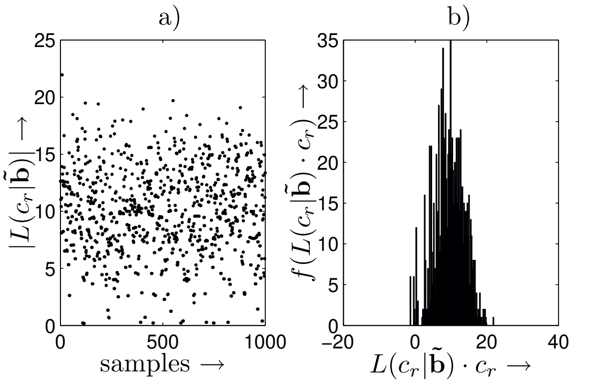


Fig. 3. Distribution of output LLRs over time (a) and histogram of conditioned LLRs (b) for $\mathcal{C}_r = [5]_8$, $\text{SNR}_{sr} = 0\,\text{dB}$, 1000 samples

show that these effects vanish for high $\text{SNR}_{sr}$. In the case of the non-recursive code $\mathcal{C}_r = [5]_8$ the output LLRs are approximately Gaussian distributed and have no obvious dependency in time direction as shown in Fig. 3. Due to this behavior it is worth to investigate both recursive and non-recursive component codes $\mathcal{C}_r$ for application in the relay.

Nevertheless, in the sequel the LLRs at the encoder output $\mathcal{C}_r$ are assumed to be Gaussian distributed and are modeled by (similar to the modeling of the decoder output suggested in [7])

$$L(c_r) = \sigma_a^2/2 \cdot c_r + n_a \quad (11)$$

with $n_a$ denoting a Gaussian random variable with zero mean and variance $\sigma_a^2$. As shown, this assumption is valid for non-recursive codes in the relay, but not for recursive codes. However, as no better assumption about the distribution is known up to now we use a Gaussian approximation as applied in [3].

## 3.2 Calculation of LLRs at the destination

For the iterative decoding the destination has to calculate LLRs for $c$ and $c_r$ on basis of the received signals $\mathbf{y}_{sd}$ and $\mathbf{y}_{rd}$. As the transmitter sends BPSK signals the corresponding LLRs for $c$ are easily calculated by

$$L(c|y_{sd}) = 2/\sigma_{sd}^2 \cdot y_{sd} \; , \tag{12}$$

where $\sqrt{G_{sd}}h_{sd} = 1$ was assumed for simplicity. In contrast the signal $\tilde{c}_r$ transmitted by the relay is continuously distributed in the range $[-1, +1]$ and therefore the calculation of LLRs is more complicated.

The total error $\eta$ of the received signal, i.e. the difference between $y_{rd}$ and the correct bit $c_r$, can be calculated as (assuming $\sqrt{G_{rd}}h_{rd} = 1$ again)

$$\eta = y_{rd} - c_r = n_{rd} - c_r\,(1 - \tilde{c}_r c_r) = n_{rd} - c_r \bar{n} \; . \tag{13}$$

In this context $c_r$ denotes the *correct* code bit assuming error-free decoding and hard output encoding using code $\mathcal{C}_r$. As the second part $c_r\bar{n}$ of (13) is not Gaussian, the total error $\eta$ is not Gaussian as well. Consequently, the simple calculation of LLRs $L(c_r|y_{rd})$ at the destination similar to (12) is not valid.

To calculate true log-likelihood values for these signals, the distribution of the noisy soft bits $y_{rd}$ is required. With the Gaussian model (11) for the LLRs the conditional distribution of the soft bits becomes

$$p(\tilde{c}_r|c_r = \pm 1) = \frac{1}{\sqrt{2\pi\sigma_a^2}} \cdot \exp\left( - \frac{\left|2\mathrm{atanh}(\tilde{c}_r) \mp \sigma_a^2/2\right|^2}{2\sigma_a^2} \right) \cdot \frac{2}{1 - \tilde{c}_r^2} \; , \tag{14}$$

which is the transformation of the Gaussian distribution of the LLRs (11) with the $\tanh(x/2)$ function [8]. To get the desired distribution of the noisy soft bits $y_{rd}$ the Gaussian distribution of the channel noise

$$p(n_{rd}) = \frac{1}{\sqrt{2\pi}\sigma_{rd}} \exp\left( - \frac{n_{rd}^2}{2\sigma_{rd}^2} \right) \tag{15}$$

and the distribution of the soft bits have to be convolved

$$p(y_{rd}|c = \pm 1) = p(\tilde{c}|c = \pm 1) \star p(n_{rd}) \; . \tag{16}$$

This convolution cannot be solved in closed form and therefore has to be done numerically. Using this result the true LLR can be calculated at the destination

$$\begin{aligned} L(c_r|y_{rd}) &= \ln\left( \frac{p(c_r = +1|y_{rd})}{p(c_r = -1|y_{rd})} \right) \\ &= \ln\left( \frac{p(y_{rd}|c_r = +1)}{p(y_{rd}|c_r = -1)} \right) + \ln\left( \frac{p(c_r = +1)}{p(c_r = -1)} \right) \; , \end{aligned} \tag{17}$$

where the second part represents a-priori information.

In contrast to this derivation, a Gaussian distribution of the total error $\eta = n_{rd} - c_r\bar{n}$ with variance $\sigma_\eta^2 = \sigma_{rd}^2 + \sigma_{\bar{n}}^2$ was assumed in [3]. Under this assumption the LLRs are approximated similar to (12)

$$L_{\text{approx.}}(c_r|y_{rd}) = 2\, \frac{(1 - \mu_{\bar{n}})^2}{\sigma_{\bar{n}}^2 + \sigma_{rd}^2} \cdot y_{rd} \; . \tag{18}$$

The variance of the desired signal part is $(1 - \mu_{\bar{n}})^2$, where

$$\mu_{\bar{n}} = E\{1 - \tilde{c}_r \cdot c_r\} \tag{19}$$

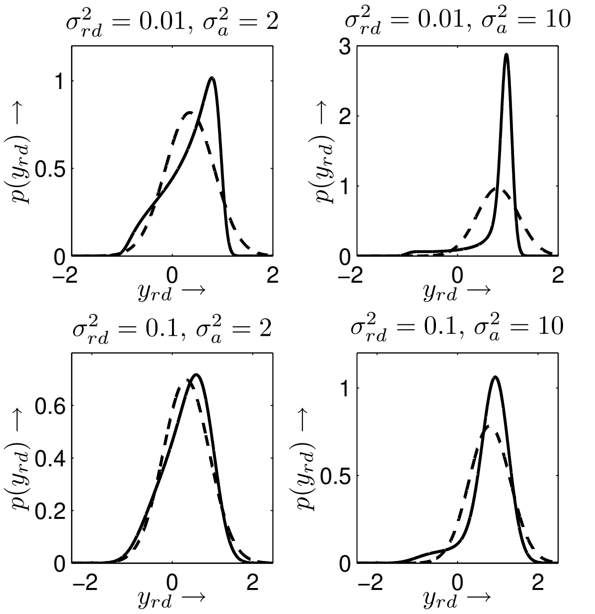


Fig. 4. Conditional PDF of received signal $p(y_{rd}|c = +1)$ with (- -) and without (–) Gaussian approximation of $\bar{n}$, different values of $\sigma_{rd}^2$ and $\sigma_a^2$

is the mean value of $\bar{n} = 1 - \tilde{c}_r \cdot c_r$ and the variance is given by

$$\sigma_{\bar{n}}^2 = E\{(1 - \tilde{c}_r \cdot c_r - \mu_{\bar{n}})^2\} \; . \tag{20}$$

Fig. 4 shows the exact (16) and the Gaussian approximated conditional distributions of the noisy soft bits used for the two approaches (17) and (18) for calculating LLRs, respectively. The difference between these distributions becomes obvious especially for low noise on the relay-destination link.

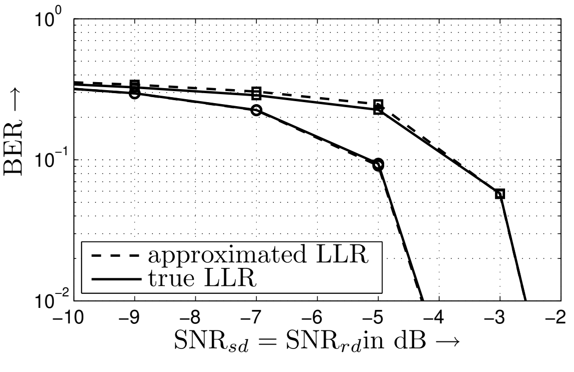


Fig. 5. BER of a distributed code $[4/7,\ 7/5;\ 1]_8$ with Gaussian approximation and exact LLRs, AWGN channel, $\mathrm{SNR}_{sr} = -3$dB (squares) and $\mathrm{SNR}_{sr} = 0\,$dB (circles), 10 iterations at destination

In Fig. 5 bit error rates (BER) using the true LLRs (17) and the approximated LLRs (18) are depicted. The Gaussian approximation of $\eta$ causes only marginal loss at low error rates. Only at higher error rates $> 10^{-1}$ the gain of the pdf-based calculation can be seen. Overall, the Gaussian approximation of $\bar{n}$ is sufficient for our purpose and as a consequence, all following results will use this approximation.

## 4 Performance Evaluation

To ensure a fair comparison, all distributed coding schemes considered in the sequel apply constituent codes of constraint length three achieving code rates of $1/2$ and $1$ at source and relay, respectively.

**a) Distributed turbo code: DTC**

The coding scheme considered in [3] consists of a systematic $[1,\, 5/7]_8$ convolutional code at the source and a $[5/7]_8$ code at the relay leading to the overall code $[1,\, 5/7;\, 5/7]_8$. This scheme is a derivation of a standard turbo code and will therefore be denoted as DTC.

Based on the fact that a recursive code in the relay suffers from error propagation and delivers non-Gaussian output LLRs, two additional schemes with non-recursive codes in the relay are considered. Consequently, the reliability of a code bit depends then only on the reliability of the information bits within the constraint length of the code.

**b) Forward systematic bits: SYS**

A simple approach is to transmit the soft information bits directly without reencoding ($\mathcal{C}_r = 1$). Consequently, two constituent codes have to be applied at the source in order to achieve a turbo-coding structure. However, as shown in [9] a code like $\mathcal{C}_s = [5/7,\, 5/7]_8$ can not be decoded without additional a-priori information, which is the case at the relay. In order to ensure decodeability at the relay we propose to use the code $\mathcal{C}_s = [4/7,\, 7/5]_8$, so that the resulting distributed code $[4/7,\, 7/5;\, 1]_8$ consists of a non-systematic code at the source and no coding at the relay, which is called SYS in the sequel. This choice of constituent codes at the source enables a convergence at high SNRs even without additional a-priori knowledge about the information bits. However, the information from the relay improves the convergence behavior especially in the case of low SNR on the source-destination hop. For this coding scheme, the relay has to decode a concatenated code and therefore, the decoding complexity in the relay is higher than for the DTC, but no soft-reencoding is required.

**c) Non-recursive code: NREC**

A tradeoff between performance and complexity may be achieved by the $[1,\, 5/7;\, 5]_8$ coding scheme with a systematic, recursive convolutional code at the source and a non-recursive code at the relay. This scheme denoted as NREC requires less decoder complexity at the relay but a soft information reencoder similar to DTC. The main drawback of this scheme is the bad asymptotic behavior. When the relay is able to decode successfully with high reliability, the performance of this coding structure is worse than that of a standard turbo-code [9].

**d) Simulation results**

Two uplink scenarios are considered representing two typical relay networks. The first assumes fixed relay stations which are located e.g. at a roof top or on a hill to ensure a good channel to the destination (e.g. base station). In this case no fading is considered on the relay-destination link. Due to the mobility of the source, e.g. mobile phone, the two links between source and relay (s-r) and between source and destination (s-d) suffer from fading, which are modeled here as independent block fading channels. The second scenario assumes a mobile relay which additionally suffers from a fading link to the destination. In this case all links are modeled as independent block fading.

In Fig. 6 the BER for a scenario with fixed relay station can be seen. The average SNR on the link from source to destination is the same as between source and relay, i.e. $\overline{\mathrm{SNR}}_{sd} = \overline{\mathrm{SNR}}_{sr}$, and the SNR between relay and destination is equal to $\mathrm{SNR}_{rd} = 7$ dB. In all cases, 10 decoding iterations are done at the destination. The NREC approach seems to have the best performance in terms of average BER and the DTC is worst. But considering frame error rates (FER) in Fig. 7, the situation changes. The SYS approach clearly outperforms the other schemes and the NREC performs worst. Additionally we depicted the FER for SYS when only 2 iterations are performed at the relay. Obviously, the performance is still superior to the other schemes with almost the same complexity.

The discrepancy between bit and frame error rates can be explained by the bit error patterns within a

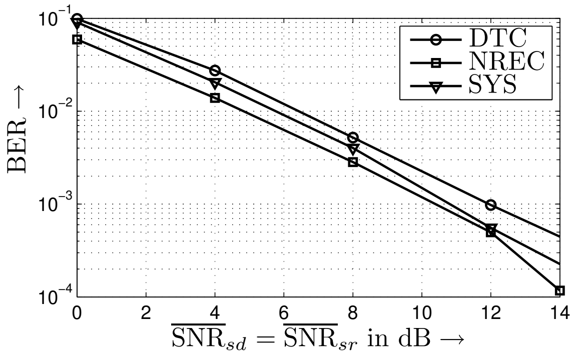


Fig. 6. Bit error rates with fixed relay, $\mathrm{SNR}_{rd} = 7$ dB

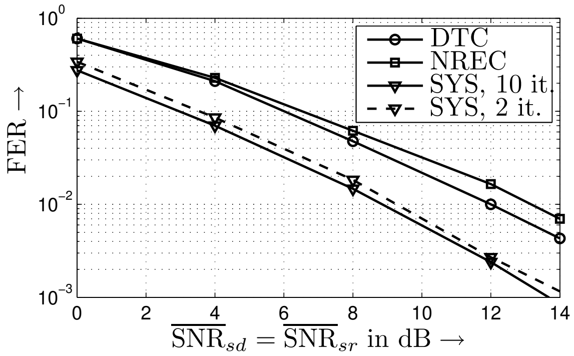


Fig. 7. Frame error rates with fixed relay, $\mathrm{SNR}_{rd} = 7$ dB

frame after reencoding at the relay. For the DTC as well as for the NREC approach only weak codes are available for error correction at the relay. In the case of DTC, few unreliable or erroneous estimates lead to a low reliability or even high error density in the whole frame which disturbs the overall decoding in the destination. The NREC scheme does not suffer from this error propagation due to the non-recursive code structure. For that reason the BER of the DTC is worse than for NREC. On the other hand, the DTC has a better asymptotic error correction capability at the destination and therefore compensates the error propagation in parts. But in general, the bit error density within one frame is higher for the DTC scheme than for NREC.

In contrast to this, the SYS approach combines the best out of both schemes: The asymptotic performance is very good and leads to a better error rate already in the relay due to turbo decoding. Additionally it avoids the error propagation by dropping the reencoding and therefore the frame error rate significantly outperforms the other schemes.

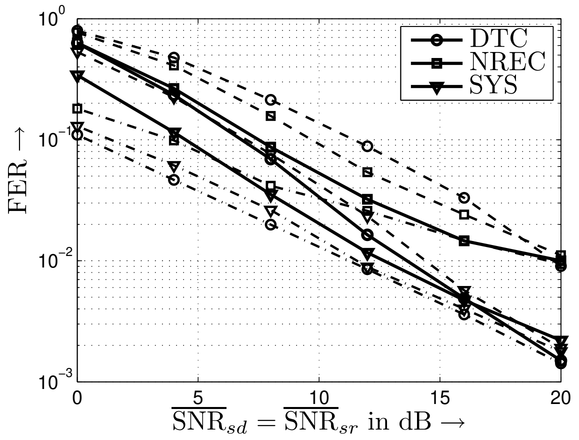


Fig. 8. Frame error rates with mobile relay, (–) soft encoded, (- -) hard encoded and (- ·) genie encoding at relay, $\overline{\mathrm{SNR}}_{rd} = 7$ dB

In wireless networks with mobile users it is possible to use other mobiles as relays. This would render fixed relay stations superfluous but suffers from worse channel properties. In this case block fading is considered on all links, even for the relay-destination link, but the average SNR is still 7dB. The results for the FER in Fig. 8 are overall worse due to fading, but the relative performance between the schemes is quite similar. Only for higher values of $\overline{\mathrm{SNR}}_{sd}$ and $\overline{\mathrm{SNR}}_{sr}$ the DTC gets better and even outperforms SYS whereas NREC gets into an error floor. In addition to the results for soft-reencoding, also error-prone decoding with hard reencoding as well as *genie encoding* assuming perfect knowledge of the information bits at the relay are shown. The soft reencoded schemes lie all in between the corresponding two extremes and tend especially at high SNR to the genie case.

# 5 Conclusions

In this paper different distributed coding schemes were compared with respect to soft-reencoding. The turbo decoder at the destination requires log-likelihood values for the signal received from the relay consisting of noisy soft bits. The exact distribution of the noisy soft bits was derived for Gaussian distributed LLRs at the reencoder output of the relay enabling the exact calculation of the received LLRs. However, it was shown via simulations that the loss due to the Gaussian approximation is negligible in the considered cases. Therefore, it seems advisable to use this approach as it leads to a much simpler calculation.

Depending on the code and its distribution in the network, the approaches have different complexities and performances. The best coding structure considered in this paper was the SYS scheme, where the source uses a non-systematic turbo-like code of rate 1/2 outperforming the schemes in [2] and [4]. The relay decodes the frames iteratively and relays the expectation value of the systematic (information) bits, which are used at the destination as additional a-priori information. This way of distributing a concatenated code over the network seems to be a good choice as well for mobile as for fixed relays in low and medium SNR regions.
The codes regarded here are only a small sample of possible distributed codes for relay networks. In the future a systematic search for appropriate constituent codes for relaying networks using soft-reencoding will be done. Furthermore, extensions with respect to adaptive relaying protocols are intended.

# References


[1] J.N. Laneman, D.N.C. Tse, and G.W. Wornell. Cooperative Diversity in Wireless Networks: Efficient Protocols and Outage Behaviour. IEEE Transactions on Information Theory, vol. 50, pp. 3062-3080, December 2004.

[2] B. Zhao and M.C. Valenti. Distributed Turbo Codes: Towards the Capacity of the Relay Channel. Proc. VTC Fall 2003, Orlando, USA, Oct. 2003.

[3] Y. Li, B. Vucetic, T. Wong and M. Dohler. Distributed Turbo Coding With Soft Information Relaying in Multihop Relay Networks. IEEE Journal on Selected Areas in Communications. Vol. 24, No. 11, pp. 2040-2050, Nov. 2006.

[4] H. Sneessens and L. Vandendorpe. Soft Decode and Forward Improves Cooperative Communications. Proc. 6th IEE International Conference on 3G and Beyond, London, UK, Nov. 2005.

[5] S. Yang and R. Koetter. Network Coding over a Noisy Relay : a Belief Propagation Approach. Proc. IEEE International Symposium on Information Theory, Nice, France, June 2007.

[6] L.R. Bahl, J. Cocke, F. Jelinek and J. Raviv. Optimal Decoding of Linear Codes for Minimizing Symbol Error Rate. IEEE Transactions on Information Theory, vol. 20, pp. 284-287, March 1974.

[7] S. tenBrink. Convergence Behavior of Iteratively Decoded Parallel Concatenated Codes. IEEE Transactions on Communications, vol. 49, pp. 1727-1737, Oct. 2001.

[8] A. Papoulis. Probability, Random Variables, and Stochastic Processes. 3rd Edition, McGraw Hill, New York, 1991.

[9] F. Brännström. Convergence Analysis and Design of Multiple Concatenated Codes. Ph.D. Thesis, Chalmers University of Technology, Göteborg, Sweden, March 2004.