Comparison of Cooperative Schemes using Joint Channel Coding and High-order Modulation

Ioannis Chatzigeorgiou, Weisi Guo, Ian J. Wassell
Digital Technology Group, Computer Laboratory
University of Cambridge, United Kingdom
Email: {ic231, wg209, ijw24}@cam.ac.uk

Rolando Carrasco
School of EE&C Engineering
University of Newcastle, United Kingdom
Email: r.carrasco@newcastle.ac.uk

***Abstract*—In this paper, we compare cooperative networks using either coded amplify-and-forward, coded cooperation or distributed turbo coding, over quasi-static fading channels under the condition of identical data rate and power consumption. We demonstrate that, when the quality of the communication channels is sufficiently high, cooperative transmission provides gains with respect to direct transmission while coded amplify-and-forward performs similarly to coded cooperation. Furthermore, we compare selfish and unselfish coded cooperation and show that, in most cases, selfishness provides a performance advantage over unselfishness.**

***Index Terms*—User cooperation, relay channel, quasi-static fading, coding, modulation, wireless network.**

I. Introduction

In wireless communication networks, multipath fading is a severe impairment which can be mitigated through the use of diversity. Space diversity based on the use of multiple antennas can be readily combined with time and frequency diversity; however, the terminals - which we call *users* or *nodes* in the paper - may not be able to support multiple antennas due to physical constraints. An alternative means of transmit diversity is *user cooperation* [1], according to which each user "overhears" its partners and relays their data to the destination.

The fundamental idea behind cooperative transmission can be traced back to the work of Cover and El Gamal [2]. Sendonaris *et al.* [1] presented a practical implementation of a user cooperation protocol employing code-division multiple access (CDMA), whilst Laneman and Wornell [3], [4] proposed several cooperative transmission schemes such as the amplify-and-forward and decode-and-forward protocols. It was shown [3], [4] that although the amplify-and-forward method achieves full diversity, i.e., its diversity order is proportional to the number of cooperating users, the decode-and-forward protocol has a fixed diversity of one. Hunter and Nosratinia [5] proposed an alternative framework, in which cooperation is integrated into channel coding; the so-called *coded cooperation* was shown to enjoy full diversity in slow fading channels.

The motivation for this paper is to compare the performance of full-diversity coded cooperative schemes in a slow fading environment with that of direct transmission. In contrast to [6], we have configured the nodes in each cooperative scenario such that the information bit rate and the power consumption per user are the same as in direct transmission; moreover, each user always transmits a fixed amount of information for its partner, independently of the selected collaborative protocol. We consider networks using the amplify-and-forward protocol [4] with convolutional coding or coded cooperation based either on convolutional coding [5] or on distributed turbo coding [7], [8]. Note that channel coding has been combined with high spectral efficiency modulation schemes, such as quadrature phase-shift keying (QPSK) and 16-point quadrature amplitude modulation (16-QAM).

The rest of this paper is organized as follows: Section II introduces the system and channel models and gives a brief description of the cooperative process. Section III presents the cooperative protocols under consideration. A performance comparison based on simulation results between cooperative schemes and direct transmission, under the condition of identical data rate and available energy per user, is carried out in Section IV. In Section V we discuss several practical issues and in Section VI we conclude and summarize the findings of our work.

II. System Model

We consider a wireless network that consists of three nodes; nodes 1 and 2 represent users transmitting to the same destination node d. The four channels between the nodes, namely user 1 to destination, user 2 to destination, user 1 to user 2 and user 2 to user 1, are subject to frequency-flat Rayleigh fading and additive white Gaussian noise. Note that the first two channels are usually referred to as *uplink channels*, whilst the other two channels are known as *inter-user channels*. Users transmit on orthogonal channels, which allows the destination to detect each user separately. Cooperation of the users occurs in two successive stages, each of which occupies a time slot. Quasi-static fading is considered, hence each channel realization remains constant for the duration of the two-stage process but changes independently from process to process.

In the remainder of the paper, we use the notation $t = 2n$ and $t = 2n + 1$ to refer to the first and second stage of the cooperation process respectively, during which the n-th channel realization is constant, where n is a nonnegative integer. During the first stage of the cooperative process, when $t = 2n$, user i, $i = 1, 2$, broadcasts its own frame $\mathbf{x}_i(2n)$ of length N_1 symbols to user j, $j = 1, 2$ with $j \neq i$, and the destination d. The baseband-equivalent discrete-time receive sequences at user j and the destination d are given by

$$\mathbf{y}_{i,j}(2n) = \sqrt{\mathcal{E}_\mathrm{s}}\, h_{i,j}(n)\, \mathbf{x}_i(2n) + \mathbf{z}_{i,j}(2n), \tag{1}$$
$$\mathbf{y}_{i,\mathrm{d}}(2n) = \sqrt{\mathcal{E}_\mathrm{s}}\, h_{i,\mathrm{d}}(n)\, \mathbf{x}_i(2n) + \mathbf{z}_{i,\mathrm{d}}(2n), \tag{2}$$

respectively. During the second stage, when $t = 2n+1$, user j transmits for the partner user i a sequence of length N_2 to the destination; this sequence, denoted as $\mathbf{x}_j(2n+1)$, has been obtained from the receive sequence $\mathbf{y}_{i,j}(2n)$ using one of the available cooperation protocols. At the end of the second stage, the receive sequence at the destination d is

$$\mathbf{y}_{j,\mathrm{d}}(2n+1) = \sqrt{\mathcal{E}_\mathrm{s}}\, h_{j,\mathrm{d}}(n)\, \mathbf{x}_j(2n+1) + \mathbf{z}_{j,\mathrm{d}}(2n+1). \tag{3}$$

When the two-stage cooperation process is completed, the destination combines the sequence received directly from user i with the sequence received through user j, according to the expression

$$\mathbf{r}_{i,\mathrm{d}}(n) = f\big(\mathbf{y}_{i,\mathrm{d}}(2n), \mathbf{y}_{j,\mathrm{d}}(2n+1)\big), \tag{4}$$

where the function $f(.)$ depends on the adopted cooperation protocol. An illustration of the cooperation process is presented in Fig.1.

In equations (1), (2) and (3), the fading coefficients $h_{1,\mathrm{d}}(n)$, $h_{2,\mathrm{d}}(n)$, $h_{1,2}(n)$ and $h_{2,1}(n)$ have been modeled as zero-mean, mutually independent, complex Gaussian random variables with variances $\sigma^2_{1,\mathrm{d}}$, $\sigma^2_{2,\mathrm{d}}$, $\sigma^2_{1,2}$ and $\sigma^2_{2,1}$, respectively. The additive noises $\mathbf{z}_{1,\mathrm{d}}(t)$, $\mathbf{z}_{2,\mathrm{d}}(t)$, $\mathbf{z}_{1,2}(t)$ and $\mathbf{z}_{2,1}(t)$ are zero-mean, mutually independent, complex Gaussian sequences with variance $\mathcal{N}_0$. The available energy per transmit symbol at user 1 and user 2 is represented by $\mathcal{E}_\mathrm{s}$, whilst the signal constellation has been normalized to unit energy.

The quality of a channel in our system model is characterized by its corresponding average receive signal-to-noise ratio (SNR). In particular, if $\gamma_{i,\mathrm{d}}(n) = |h_{i,\mathrm{d}}(n)|^2 \mathcal{E}_\mathrm{s}/\mathcal{N}_0$ is the instantaneous receive SNR for an uplink channel, the average receive SNR, $\overline{\gamma_{i,\mathrm{d}}}$, for the same channel assumes the form

$$\overline{\gamma_{i,\mathrm{d}}} = \mathbb{E}\big[\gamma_{i,\mathrm{d}}(n)\big] = \mathbb{E}\Big[\big|h_{i,\mathrm{d}}(n)\big|^2\Big]\frac{\mathcal{E}_\mathrm{s}}{\mathcal{N}_0} = \sigma^2_{i,\mathrm{d}}\frac{\mathcal{E}_\mathrm{s}}{\mathcal{N}_0}, \tag{5}$$

where $\mathbb{E}[.]$ denotes the expectation operator and $\sigma^2_{i,\mathrm{d}} \triangleq \mathbb{E}\big[|h_{i,\mathrm{d}}(n)|^2\big]$, since $h_{i,\mathrm{d}}(n)$ is a zero-mean random variable. The average receive SNR, $\overline{\gamma_{i,j}}$, for the inter-user channels can be obtained in a similar fashion. When the average receive SNRs for the two uplink channels are equal, the channels are referred to as *statistically similar* [5], whilst the corresponding network of nodes is called *symmetric* [9].

Throughout this paper, we assume that channel state information at the receivers is available and coherent detection is possible. For simplicity, we also assume that the inter-user channels are reciprocal, so that $h_{1,2}(n) = h_{2,1}(n)$. Finally, the transmit sequences $\mathbf{x}_i(2n)$ and $\mathbf{x}_j(2n+1)$ are taken to have equal length N, i.e., $N_1 = N_2 = N$. Consequently, the *level of cooperation* between the two users, defined as the percentage of symbols transmitted by a user for its partner during the two-stage process [5], is set to $N_2/(N_1+N_2) = 0.5$.

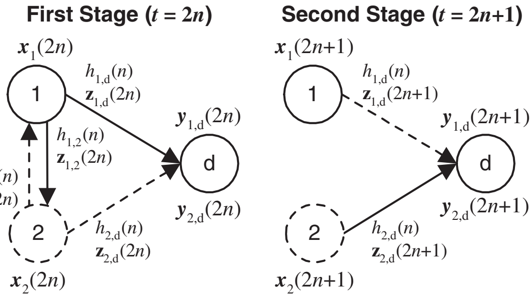

Fig. 1. System model for cooperative transmission

III. Cooperative Transmission Protocols

In this section, we describe three cooperation protocols, namely coded amplify-and-forward, coded cooperation and distributed turbo coding. A direct transmission scheme is used as a reference when comparing the performance of the cooperative protocols under investigation.

A. Direct Transmission

In direct transmission, we assume that both users employ a coding scheme of rate ρ and a modulation scheme of order $\mathcal{M}_\mathrm{D}$. At time slot $t = n$, user i transmits a sequence $\mathbf{x}_i(t)$ and the destination d receives

$$\mathbf{r}_{i,\mathrm{d}}(n) = \mathbf{y}_{i,\mathrm{d}}(n) = \sqrt{\mathcal{E}_\mathrm{s}}\, h_{i,\mathrm{d}}(n)\, \mathbf{x}_i(n) + \mathbf{z}_{i,\mathrm{d}}(n). \tag{6}$$

Direct transmission is equivalent to a non-cooperative scheme, according to which a user broadcasts half of its source data during the first stage, whilst the same user relays no symbols for its partner during the second stage but instead transmits its own remaining symbols to the destination. In a cooperative scheme however, a user can transmit its own data only in the first stage of cooperation. Therefore, each cooperating user would have to increase the cardinality of the modulation scheme from $\mathcal{M}_\mathrm{D}$ to $\mathcal{M}^2_\mathrm{D}$ so as to double its throughput and thus maintain the same information bit rate as in direct transmission. Furthermore, if our objective is to keep the energy consumption constant in both direct and cooperative scenarios, the available energy per source bit at each user should be half of that allocated in direct transmission.

B. Coded Amplify-and-Forward

When the coded amplify-and-forward (AF) protocol [4] is employed in the second stage of cooperation, each user dedicates all its bandwidth to relay to the destination an amplified version of its partner's transmit sequence. In particular, in the first stage of the process user j receives a sequence $\mathbf{y}_{i,j}(2n)$ from user i, as described in (1); in the second stage user j relays the receive sequence by transmitting $\mathbf{x}_j(2n+1) = \beta\mathbf{y}_{i,j}(2n)$ to the destination d. The amplifying gain β is set to

$$\beta = \Big(\big|h_{i,j}(n)\big|^2\mathcal{E}_\mathrm{s} + \mathcal{N}_0\Big)^{-1/2}, \tag{7}$$

such that the available transmit energy per symbol at user j is scaled to $\mathcal{E}_{\mathrm{s}}$ [4]. Using (2) and (3), we can compute user's i direct and relayed copies of the transmit sequence, namely $\mathbf{y}_{i,\mathrm{d}}(2n)$ and $\mathbf{y}_{j,\mathrm{d}}(2n+1)$, which were received by the destination d at the end of the first and second stage of the process, respectively. These two independent copies are optimally combined prior to de-mapping, as follows [4]

$$\mathbf{r}_{i,\mathrm{d}}(n)=h^*_{i,\mathrm{d}}(n)\mathbf{y}_{i,\mathrm{d}}(2n)+\frac{h^*_{i,j}(n)\beta h^*_{j,\mathrm{d}}(n)}{\beta^2|h_{j,\mathrm{d}}(n)|^2+1}\,\mathbf{y}_{j,\mathrm{d}}(2n+1), \quad (8)$$

where the notation ξ^* is used to denote the conjugate of a complex number ξ. Note that knowledge of the amplifying gain β as well as the fading coefficients of all communication links is required at the destination.

C. Coded Cooperation

In coded cooperation [5], each user employs two codes, $\mathcal{C}_1$ and $\mathcal{C}_2$, one for each stage of the cooperative process. Both codes have the the same number of inputs, the same memory size and the same rate ρ. In the first stage of the process, user j receives a sequence $\mathbf{y}_{i,j}(2n)$ from user i and decodes it using a decoder for the rate-ρ code $\mathcal{C}_1$. In the second stage, user j re-encodes the information bits of its partner using $\mathcal{C}_2$, modulates the parity check bits and transmits a sequence $\mathbf{x}_j(2n+1)$ to the destination. If both users have successfully decoded one another's source data, the destination coherently detects and concatenates the sequence received directly from user i with the sequence received from the partner user j, as follows

$$\mathbf{r}_{i,\mathrm{d}}(n) = \{h^*_{i,\mathrm{d}}(n)\mathbf{y}_{i,\mathrm{d}}(2n),\; h^*_{j,\mathrm{d}}(n)\mathbf{y}_{j,\mathrm{d}}(2n+1)\}. \quad (9)$$

Recall that $\mathbf{y}_{i,\mathrm{d}}(2n)$ consists of codewords generated by $\mathcal{C}_1$, whilst $\mathbf{y}_{j,\mathrm{d}}(2n+1)$ contains codewords generated by $\mathcal{C}_2$. Demodulation of the concatenated sequence $\mathbf{r}_{i,\mathrm{d}}(n)$ of $2N$ symbols generates a sequence of soft-valued parity check bits; those bits are multiplexed so as to form codewords of a rate-$\rho/2$ code, denoted as $(\mathcal{C}_1, \mathcal{C}_2)$, which is the parallel concatenation of $\mathcal{C}_1$ and $\mathcal{C}_2$. The destination can then employ a convolutional decoder to recover the source bits of user i.

Nevertheless, user j may not successfully decode the sequence of its partner i at the end of the first stage. According to [9], user j could perform error detection along with error correction. If user j cannot decode its partner's sequence correctly, it notifies user i before the beginning of the second stage of the cooperative process. In that case, cooperation is aborted and both users employ $\mathcal{C}_2$ to generate additional coded symbols for their own source data, which they transmit to the destination in the second time slot. Consequently, the destination coherently detects and concatenates the two sequences received directly from user i (or j), i.e.,

$$\mathbf{r}_{i,\mathrm{d}}(n) = \{h^*_{i,\mathrm{d}}(n)\mathbf{y}_{i,\mathrm{d}}(2n),\; h^*_{i,\mathrm{d}}(n)\mathbf{y}_{i,\mathrm{d}}(2n+1)\}, \quad (10)$$

and uses a decoder for the overall rate-$\rho/2$ code $(\mathcal{C}_1, \mathcal{C}_2)$ to retrieve the source bits of user i. We refer to this form of cooperation, according to which either both users cooperate or do not cooperate, as *selfish* coded cooperation.

A different approach has been proposed in [5]; user j could act independently in the second stage of the process, with no knowledge of whether its own sequence has been successfully decoded by the partner user i. Hence, if user j correctly decodes user's i sequence but user i fails to decode user's j sequence, both users will use $\mathcal{C}_2$ in the second time slot to generate coded symbols for user's i source data. Thus, at the end of the two-stage process, the destination concatenates the sequence $\mathbf{y}_{i,\mathrm{d}}(2n)$, which has received directly from user i at the end of the first stage, with the optimally combined sequences $\mathbf{y}_{i,\mathrm{d}}(2n+1)$ and $\mathbf{y}_{j,\mathrm{d}}(2n+1)$, which have been simultaneously received at the end of the second stage. The outcome

$$\begin{aligned}\mathbf{r}_{i,\mathrm{d}}(n) = \{&h^*_{i,\mathrm{d}}(n)\mathbf{y}_{i,\mathrm{d}}(2n),\\ &h^*_{i,\mathrm{d}}(n)\mathbf{y}_{i,\mathrm{d}}(2n+1) + h^*_{j,\mathrm{d}}(n)\mathbf{y}_{j,\mathrm{d}}(2n+1)\},\end{aligned} \quad (11)$$

is then demodulated, multiplexed and decoded by a rate-$\rho/2$ decoder for $(\mathcal{C}_1, \mathcal{C}_2)$. We note that in the afore-mentioned case, user j transmits its own coded sequence during the first stage. The receive sequence at the destination

$$\mathbf{r}_{j,\mathrm{d}}(n) = h^*_{j,\mathrm{d}}(n)\mathbf{y}_{j,\mathrm{d}}(2n), \quad (12)$$

is demodulated and decoded using a rate-ρ decoder for $\mathcal{C}_1$. In this form of cooperation, which we call *unselfish* coded cooperation, a user always cooperates if it has successfully retrieved its partner's source data.

D. Distributed Turbo Coding

Distributed turbo coding (DTC), also known as turbo-coded cooperation, has been simultaneously proposed by Hunter *et al.* [8], [10] and Zhao and Valenti [7], [11]. As in coded cooperation, each user employs a rate-ρ code $\mathcal{C}_1$ in the first stage of the process and a rate-ρ code $\mathcal{C}_2$ in the second stage. However, a user that has successfully retrieved the information bits of its partner, first interleaves the recovered bits and then re-encodes them using $\mathcal{C}_2$. Owning to the presence of the interleaver, the two users have cooperatively formed a rate-$\rho/2$ turbo code [12], denoted as $\mathcal{T}(\mathcal{C}_1, \mathcal{C}_2)$, which is distributed in space. Consequently, the destination can use $\mathcal{C}_1$ and $\mathcal{C}_2$ as constituent codes of a turbo decoder to iteratively estimate the source bits of each user. Similarly to coded cooperation, DTC could be based either on a selfish or unselfish protocol.

IV. Performance Comparison

In this section, we provide simulation results for all three cooperative protocols and we compare their performance with that of direct transmission.

In direct transmission, we have concatenated the best rate-1/2 eight-state non-recursive non-systematic convolutional code, having generator polynomials (13,17) in octal form [13], with a Gray-coded QPSK modulator. In the coded AF protocol, the same channel code is used but Gray-coded 16-QAM is adopted instead of QPSK. Note that we have used 16-QAM in all coded cooperative schemes in order to maintain the information bit rate of a user equal to that of direct transmission. In coded cooperation, we have selected the generator polynomials

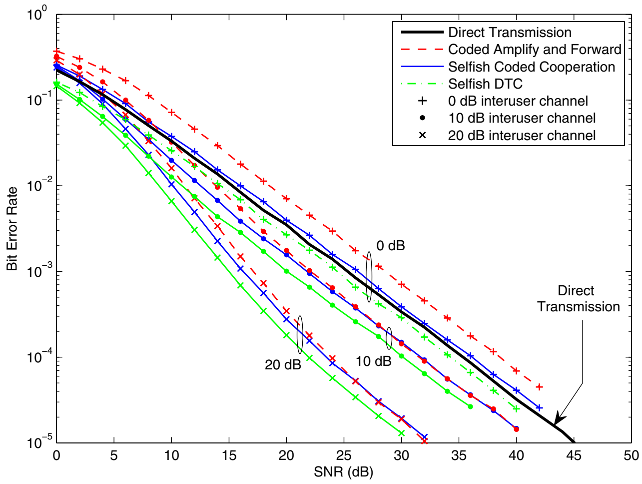

Fig. 2. Cooperation in a symmetric network

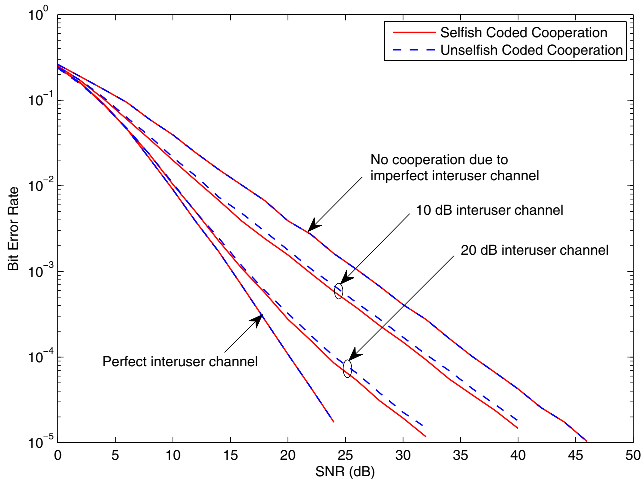

Fig. 4. Comparison of selfish and unselfish coded cooperation

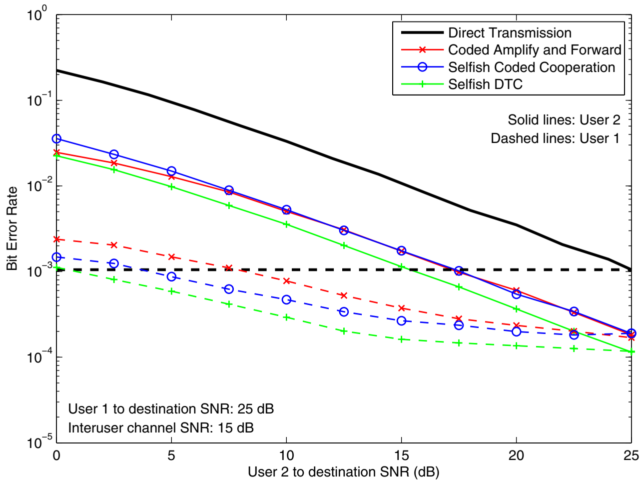

Fig. 3. Cooperation in an asymmetric network

(13,17) and (15,13) for $\mathcal{C}_1$ and $\mathcal{C}_2$, respectively; the overall rate-1/4 convolutional code (13,17,15,13) was shown to exhibit good performance in fading environments [9]. In all aforementioned scenarios, the Viterbi decoding algorithm [14] is used to retrieve the information bits. In DTC, each user employs the same rate-1/2 eight-state recursive systematic convolutional code, denoted as (1,17/13), to implement both $\mathcal{C}_1$ and $\mathcal{C}_2$; a turbo code using that particular constituent code has been reported to yield good performance [10], [15]. A user that has successfully recovered the source bits of its partner, permutes them using an S-random interleaver [16] prior to encoding. The destination uses the optimal maximum a-posteriori algorithm in the log domain (log-MAP) [17] to iteratively decode the overall rate-1/4 distributed turbo code $\mathcal{T}(1, 17/13, 1, 17/13)$; eight iterations are considered. In all cooperative transmission protocols as well as in direct transmission, the length of the source information sequence is 128 bits; furthermore, soft de-mapping [18] of the receive symbols always precedes channel decoding.

Fig. 2 shows simulation results of the various schemes, when both users have uplink channels of similar quality to the destination. As expected, the performance gain of user cooperation over direct transmission significantly increases as the quality of the reciprocal inter-user channel improves (e.g. for $\overline{\gamma_{i,j}} > 0$ dB). Interestingly, we observe that the performance of coded AF approaches that of coded cooperation as the average SNR of both the symmetric uplink channel and the inter-user channel increases; as long as the inter-user channel is error-prone, there is a point beyond which coded AF achieves a performance similar to that of coded cooperation. We attribute this to the error detection scheme embedded in coded cooperation; the small number of errors that occur even when the inter-user channel is good prevent the users from cooperating, hence the overall performance does not further improve. On the contrary, a node using the coded AF protocol forwards those errors to the destination, which can successfully correct them if the average SNR of the uplink channel is sufficiently high.

In Fig. 3 we compare the performance of the schemes under investigation, when the users have dissimilar uplink channels. In particular, the average uplink SNR for user 1 is fixed at 25 dB, the average uplink SNR for user 2 varies from 0 dB to 25 dB and the SNR of the inter-user channel is set to 15 dB. We observe that the performance of user 2, which experiences the worse uplink channel, improves markedly by cooperating; cooperation can also be beneficial for user 1, depending on the quality of its partner's uplink channel and the adopted protocol. Note that DTC achieves the best performance in both symmetric and asymmetric scenarios.

Both Fig. 2 and 3 depict the performance of selfish cooperation. In Fig. 4, we compare it to that of unselfish cooperation. Both approaches result in the same performance when the inter-user channel is either very poor or perfect; in the former case the users never cooperate whilst in the latter case they always cooperate. In any other case, the overall system performance is determined by the strategy followed when only one of the users has successfully decoded its partner's source data. We see that the system performance is better when that user decides to transmit additional parity check bits for its own source data (selfish cooperation) than transmit additional information for its partner (unselfish cooperation).

V. Discussion

In the previous section we established that user cooperation can provide a performance gain over direct transmission. Furthermore, among the three candidate protocols, DTC achieved the best error rate performance, followed by coded cooperation and coded AF. Nevertheless, the performance of coded AF is comparable to that of coded cooperation, if the uplink channel SNR and inter-user channel SNR are sufficiently high. In this section, we discuss some implementation issues for each collaborative protocol.

Coded AF has a low computational complexity since the partner just forwards the receive data in the analog domain and the destination employs a high rate decoder to retrieve them. Nevertheless, according to (8), the destination can optimally combine the receive sequences only if it has knowledge of the amplifying gain as well as state information for all channels. Consequently, a partner is required to determine both the fading coefficient of the inter-user channel and the amplifying gain, then employ an error correction scheme to protect this information and finally transmit it to the destination. Inevitably, this additional information causes an overhead that reduces the overall rate of the system.

Computational complexity increases at both the partner and the destination if we adopt coded cooperation; the partner decodes and re-encodes the receive data while the destination uses a lower rate convolutional decoder than the one used in coded AF. In addition, the partner introduces an overhead when communicating with the destination, as in coded AF. More specifically, a user transmits in the second stage of cooperation one additional bit to the destination [5] that indicates whether the user has sent its own data or parity check information for its partner. Of course, this bit would need error protection which would affect the rate of coded cooperation.

Although a selfish implementation of coded cooperation yields a small performance advantage over the unselfish approach, the exchange of information between the partners at the end of the first stage of cooperation introduces a delay, which might have a negative impact on real-time applications. In particular, a user encodes a single bit of information [9] that indicates whether decoding of its partner's receive data was successful or not and transmits it to the partner user. Upon receiving this message and successfully decoding it, the partner decides whether to cooperate or not, accordingly.

Adoption of DTC mainly adds a small delay at the partner owning to the introduction of the bit interleaver and a significant delay as well as a considerable increase of the computational complexity at the destination due to the replacement of the convolutional decoder with the turbo decoder. For example, the turbo decoding configuration of this paper performs about 40 times more operations than the convolutional decoder used in coded cooperation, based on the complexity expressions in [19]. Consequently, DTC is more appropriate for non real-time applications, such as Internet browsing.

VI. Conclusion

In this paper we considered three full-diversity coded cooperative schemes using high-order modulation and we compared their performance to that of direct transmission, under the condition of identical data rate and available energy per user. We demonstrated that cooperation can be beneficial for both users when the quality of the inter-user and the uplink channels is relatively good and we illustrated that selfish behavior in coded cooperation provides a small performance gain over unselfish behavior. Finally, we briefly discussed the implementation issues of user cooperation and concluded that coded amplify-and-forward and unselfish coded cooperation are more appropriate for real-time communications.

Acknowledgment

The authors would like to acknowledge the financial support of the Engineering and Physical Sciences Research Council (EPSRC) under Grant EP/E012108/1.

References

[1] A. Sendonaris, E. Erkip, and B. Aazhang, "User cooperation diversity, Part I and Part II," *IEEE Trans. Commun.*, vol. 51, no. 11, pp. 1927–1948, Nov. 2003.

[2] T. M. Cover and A. E. Gamal, "Capacity theorems for the relay channel," *IEEE Trans. Inf. Theory*, vol. IT-25, pp. 572–584, Sep. 1979.

[3] J. N. Laneman and G. W. Wornell, "Energy-efficient antenna sharing and relaying for wireless networks," in *Proc. IEEE Wireless Comm. and Networking Conference*, Sep. 2000.

[4] ——, "Cooperative diversity in wireless networks: Efficient protocols and outage behavior," *IEEE Trans. Inf. Theory*, vol. 50, no. 12, pp. 3062–3080, Dec. 2004.

[5] T. Hunter and A. Nosratinia, "Diversity through coded cooperation," *IEEE Trans. Wireless Commun.*, vol. 5, pp. 283–289, Feb. 2006.

[6] A. Nosratinia, T. Hunter, and A. Hedayat, "Cooperative communication in wireless networks," *IEEE Comm. Mag.*, vol. 42, pp. 74–80, Oct. 2004.

[7] B. Zhao and M. C. Valenti, "Distributed turbo coded diversity for the relay channel," *Electron. Lett.*, vol. 39, no. 10, pp. 786–787, May 2003.

[8] T. Hunter and A. Nosratinia, "Coded cooperation with space-time transmission and iterative decoding," in *Proc. WNCG Wireless Networking Symposium*, Oct. 2003.

[9] A. Stefanov and E. Erkip, "Cooperative coding for wireless networks," *IEEE Trans. Commun.*, vol. 52, no. 9, pp. 1470–1476, Sep. 2004.

[10] M. Janani, A. Hedayat, T. Hunter, and A. Nosratinia, "Coded cooperation in wireless communications: Space-time transmission and iterative decoding," *IEEE Trans. Sig. Proc.*, vol. 52, pp. 362–371, Feb. 2004.

[11] M. Valenti and B. Zhao, "Distributed turbo codes: Towards the capacity of the relay channel," in *Proc. IEEE Vehic. Tech. Conf.*, Oct. 2003.

[12] C. Berrou and A. Glavieux, "Near optimum error correcting coding and decoding: Turbo codes," *IEEE Trans. Commun.*, vol. 44, no. 2, pp. 1261–1271, Oct. 1996.

[13] M. Bossert, *Channel Coding for Telecommunications*. John Wiley 1999.

[14] A. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE Trans. Inf. Theory*, vol. 13, no. 2, pp. 260–269, Mar. 1967.

[15] S. Benedetto, R. Garello, and G. Montorsi, "A search for good convolutional codes to be used in the construction of turbo codes," *IEEE Trans. Commun.*, vol. 46, no. 9, pp. 1101–1105, Sep. 1998.

[16] S. Dolinar and D. Divsalar, *Weight Distribution for Turbo Codes using Random and Nonrandom Permutations*. JPL Prog. Rep. 42-122, 1995.

[17] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimising symbol error rate," *IEEE Trans. Inf. Theory*, vol. IT-20, pp. 284–287, Mar. 1974.

[18] S. T. Brink, J. Speidel, and R. H. Yan, "Iterative demapping and decoding for multilevel modulation," in *Proc. IEEE Global Telecom. Conference*, Nov. 1998.

[19] I. Chatzigeorgiou, M. R. D. Rodrigues, I. J. Wassell, and R. Carrasco, "Comparison of convolutional and turbo coding for broadband FWA systems," *IEEE Trans. Broadcast.*, vol. 53, pp. 494–503, Jun. 2007.