Multiple Packet Reception in Asynchronous Wireless Networks

Antonios Argyriou*†
*Department of Computer and Communication Engineering, University of Thessaly, Volos, 38221, Greece
†The Center for Research and Technology Hellas (CERTH), Thessaloniki, 57001, Greece

***Abstract*—In this paper we present a scheme for multiple-packet reception (MPR) that allows the concurrent packet transmission from several asynchronous users and is based on a new algorithm for successive interference cancellation (SIC). Transmitted signals are superimposed at the destination nodes and also at a relay node. The system model allows for the transmitted packets to be superimposed asynchronously with an arbitrary pattern at each node that receives them. For the final version of the proposed scheme we present a joint MPR and channel decoding algorithm.**

***Index Terms*—Wireless networks, cooperative systems, multiple packet reception, physical layer network coding, successive interference cancelation**

I. Introduction

Multiple access in wireless networks is usually addressed through a contention mechanism that ensures exclusive access to the channel by a single node. When collisions occur, the collided packets are discarded and then they are retransmitted. This event increases the packet transmission delay while it also results in lower utilization of the wireless channel. The aforementioned negative viewpoint on collisions was altered with the introduction of the concept of multiple packet reception (MPR). With MPR nodes in wireless networks can decode multiple packets from collided packets that are not discarded [1]. Network-assisted diversity multiple access (NDMA) [2] is a typical MPR scheme that combines the collided packets in order to extract the original packet. More recently, colliding transmissions between cooperating wireless network nodes has been used for improving the performance of the wireless network as a whole. The technique is usually referred as physical layer network coding (PLNC) [3], [4], [5], [6]. With PLNC a wireless node transmits a packet in such a way that it interferes/collides with a second transmission, when it knows that this second transmitted packet is known at the nodes that need the primary packet. In [6] we investigated the superimposed transmission of packets only from two independent senders and a single relay that were transmitting under the complete control of a higher layer protocol. This is also the case in several related works that allow collisions [2], [4], [5].

In this paper we present a system that allows multiple users to transmit for asynchronous MPR (AMPR) and its contribution is on three levels: First, users are allowed to transmit packets of an arbitrary length. This relaxed assumption is possible both for successive packets that originate from the same user or packets transmitted from different users. Second, packet transmissions can occur at different time instants across sending users while only loose synchronization is required between them. When users desire to send a packet, their transmissions only have to be aligned across the symbol boundaries of the receiving nodes, while a packet scheduler or a MAC protocol does not need to ensure the precise time instant that the transmission will take place. Decoding of the appropriate packet at each destination is ensured with the use of a novel *two-dimensional successive interference cancellation* (2DSIC) and decoding algorithm.

The author would like to acknowledge the support from the European Commission through the Marie Curie Intra-European Fellowship WINIE-273041 and the STREP project CONECT (FP7ICT257616).

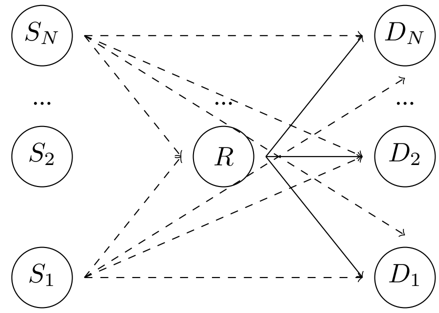


Fig. 1. The network model that we adopt in this paper includes multiple senders/users, multiple destinations, and a single relay. Different line styles indicate transmission in a different time slot. Transmissions from S_2 are not shown to avoid clogging the figure.

II. Transmission Scheme

The main idea of the transmission scheme is that all users broadcast during a number of B broadcast transmission phases and if there is an available relay, in the next transmission phase it forwards the received signals. In order to ensure linear processing at the receiver the number of required relaying (M) plus broadcast (B) phases is equal to the number of users that transmit concurrently (N). Therefore, is must be $M+B=N$. We use a toy example based on the topology depicted in Fig. 1. Fig. 2 presents the corresponding protocol behavior in the time domain. In this example there is first a broadcast phase from all the three users and one subsequent forwarding phase from the relay. Because of spatial diversity, different versions of the broadcasted signals are received at different network nodes including the relay. During the forwarding phase, the relay broadcasts the locally received superimposed signals

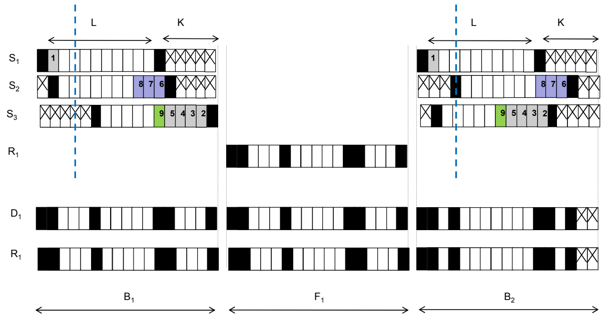


Fig. 2. Behavior of the transmission scheme with one relay. The dark-shaded blocks indicate the symbols that belong to preambles and postambles of a packet. The blocks with an X indicate the empty symbol slots for the corresponding sender. The thick vertical dashed lines depict the fourth symbol slot where misalignment can occur but only in different broadcast phases.

after it applies the appropriate power scaling. Since one relay is available, all the senders transmit again concurrently in the second and final broadcast phase as Fig. 2 indicates.

III. Two-Dimensional SIC

A. X-Dimension SIC Algorithm

The decoding algorithm described in this section is completely different from the classic notion of SIC since the proposed scheme cancels symbols that are superimposed with different symbols every time they are transmitted. The main idea of the algorithm is that cancelation is applied across multiple and inconsistently aligned symbol streams along the X dimension. Fig. 2 is used for explaining how the algorithm works. In Fig. 2 the dashed vertical lines show how different symbols, that belong to the same packets, are aligned with a different pattern in the two different transmission phases namely B_1 and B_2. If we look the second transmitted symbol from S_1, it is marked as 1 and is "clear" during phase B_1 since it is only superimposed with a known preamble from sender S_2. Therefore, this symbol can be detected on its own and it is the first to be detected. However, it cannot help for the detection of another symbol during phase B_1. Therefore, the signal received during B_1 is *parsed backwards* from right to left starting from the postamble. In this way symbols marked as 2,3,4,5 can be detected immediately because they are not superimposed with anything else. This information is stored and the algorithm proceeds to the next broadcast phase, i.e. B_2. The same process starts which allows the detection of symbols marked as 6 and 7. However, at this point the algorithm can also detect symbol marked 8 since it can cancel the symbol marked as 2. Once the algorithm cannot detect any more symbols in this way it starts by parsing again the first phase B_1. With similar logic the known symbol 6 now allows the decoding of the symbol marked as 9. When during the complete forward and backward parsing of the symbol slots of each broadcast phase does not lead to any more decoded symbols, the algorithm stops. With this logic, the X-Dimension SIC algorithm allows a different symbol alignment in every broadcast phase.

$xdim\_sic(D_k)$

1: $\hat{\mathbf{x}} \leftarrow \mathbf{0}$, $\tilde{\mathbf{x}} \leftarrow \mathbf{x}$, $\mathbf{U}_{j,t} \leftarrow \mathbf{0}$
2: **for** transmission phase $j = 1$ to N **do**
3:   **for** $t = 1$ to $L + K = T$ **do**
4:     **if** $y_j[t] > 0$ **then**
5:       **for** sender $n = 1$ to N **do**
6:         **if** $SOP(n,j)$ already identified? **then**
7:           $\mathbf{U}_{j,t}(n) = \mathbf{U}_{j,t}(n) + 1;$
8:         **else if** $EOP(n,j)$ detected **then**
9:           $\mathbf{U}_{j,t}(n) = 0$
10:         **end if**
11:       **end for**
12:       **if** $enum(\mathbf{U}_{j,t}) == 1$ **then**
13:         $\hat{\mathbf{x}}(\mathbf{U}_{j,t}(n)) = y_j[t]/h_{S_n,D_k}$
14:       **end if**
15:       $\tilde{y}_j[t] = y_j[t]$
16:       **for** sender $n = 1$ to N **do**
17:         **if** symbol $\hat{\mathbf{x}}(\mathbf{U}_{j,t}(n)) \neq 0$ (detected) **then**
18:           $\tilde{y}_j[t] = \tilde{y}_j[t] - h_{S_n,D_k} \cdot \hat{\mathbf{x}}(\mathbf{U}_{j,t}(n))$
19:           $\tilde{\mathbf{x}}(\mathbf{U}_{j,t}(n)) = 0$
20:         **end if**
21:       **end for**
22:     **end if**
23:   **end for**
24:   $ydim\_decode(\tilde{\mathbf{y}}[t], \tilde{\mathbf{x}}[t], \tilde{\mathbf{h}}_{S_n,D_k})$
25: **end for**

Fig. 3. Pseudo-code for the 2DSIC algorithm.

The pseudo-algorithm for the X-Dimension SIC can be seen in Fig. 3. This algorithm is executed after all the superimposed signals have been received at the destination nodes that desire to decode a specific packet. The pseudo-code is basically parsing symbol slots that were defined by the system parameter T from left to right (thus the name X-Dimension SIC). The first important task is to identify if in a specific symbol slot a specific user transmitted a symbol. This is done in line 4 by simply checking the energy of the received signal $y_j[t]$ during phase j and slot t. Next, if the start-of-packet (SOP) has been identified for node n and phase j ($SOP(n,j)$) and the end-of-packet (EOP) has not yet been reached, the index for the number of the received symbols is increased for that particular sender for each increase of the counter of symbol slots t. This information is kept in the vector $\mathbf{U}_{j,t}(n)$ that contains the id of the symbol that was transmitted from sender n at phase j and in time slot t. The above are performed in lines 5-11. The aforementioned process is critical since it identifies which nodes have transmitted a symbol for transmission phase j and slot t. The detection of a symbol actually happens in lines 12-14 when the contents of $\mathbf{U}_{j,t}$ are checked to identify the case that a single node has transmitted. Furthermore, the received signal is equalized by the respective channel gain h_{S_n,D_k}. If more than two users transmit concurrently, the algorithm checks in line 17 if any of these symbols have been detected in the previous symbol slots. If this is the case, then all the detected symbols are removed from the received signal (line 18). When all the transmission phases are "parsed" in this way, the decoder has collected at most N equations with maximum N unknowns for a specific symbol slot t while the algorithm has canceled along the X dimension all the previously detected symbols and are contained in the matrix of estimated symbols $\hat{\mathbf{x}}$. Also the helper matrix $\tilde{\mathbf{x}}$ contains the unknown symbols that remain to be detected after this process is finished, and is passed to the Y-dimension SIC and decoding algorithm.

B. Y-Dimension SIC and Decoding Algorithms

We now describe the detection algorithm executed at the destination node for each symbol slot t. In Fig. 2 we see that the algorithm detects symbols aligned along the Y dimension, an observation that provided it its name. It is important to stress that after the X-Dimension SIC has been applied we use the notation $\tilde{\mathbf{y}}[t]$ to denote the transmitted signal after the canceled symbols have been removed and with $\tilde{\mathbf{x}}[t]$ to denote the remaining symbols that still need detection. This means that the number of unknowns and linear equations that have to be solved is reduced. In matrix form we have again for one symbol slot t and all the phases that the received signal at D_k is:

$$\tilde{\mathbf{y}}_{D_k}[t] = \begin{bmatrix} y^{(1)}_{\mathcal{S},D_k}[t] & \ldots & y^{(B)}_{\mathcal{S},D_k}[t] & y^{(1)}_{\mathcal{S},R,D_k}[t] \ldots & y^{(M)}_{\mathcal{S},R,D_k}[t] \end{bmatrix}^T$$

For proceeding with decoding we also need to express the joint channel matrix that includes the broadcast and forwarding phases as we did with the received signals above:

$$\mathbf{H}_{D_k} = \begin{bmatrix} h_{S_1,D_k} & .. & h_{S_N,D_k} \\ h_{S_1,D_k} & .. & h_{S_N,D_k} \\ \ldots & \ldots & \ldots \\ h_{S_1,R}h_{R,D_k} & .. & h_{S_N,R}h_{R,D_k} \\ \ldots & \ldots & \ldots \end{bmatrix}$$

The matrix that corresponds to the power scaling applied by the relay during each forwarding phase m is denoted as

$$\mathbf{G} = \begin{bmatrix} g_1 & \ldots & g_m & \ldots & g_M \end{bmatrix},$$

where

$$g_m = \sqrt{\frac{P}{P\sum_{n=1}^{N}|h_{n,m}|^2 + \sigma^2}}.$$

If P is the transmit power from each sender, we can write in vector form all the received signals at destination D_k for a symbol slot t :

$$\tilde{\mathbf{y}}_{D_k}[t] = \sqrt{P}\mathbf{G}\mathbf{H}_{D_k}\tilde{\mathbf{x}}[t] + \mathbf{w}_{D_k} \tag{1}$$

Similarly with above the noise matrix is:

$$\mathbf{w}_{D_k}[t] = \begin{bmatrix} w^{(1)}_{D_k} & w^{(2)}_{D_k} & \ldots & g_R h_{R,D_k} w^{(1)}_R + w^{(1)}_{D_k} \end{bmatrix}^T$$

After we use the above description for the signals of interest we can proceed and define the decoding method for the signals that are aligned along the Y dimension.

For a multi-user system the optimal detector is an MMSE-SIC receiver [7]. The Minimum Mean Square Error (MMSE) approach tries to find a coefficient matrix $\mathbf{Q}$ which minimizes the MMSE criterion. If the Hermitian of $\mathbf{H}$ is $\mathbf{H}^H$, then the pseudo-inverse channel matrix is defined $\mathbf{H}^\dagger = (\mathbf{H}^H\mathbf{H})^{-1}\mathbf{H}^H$. But for MMSE we have that $\mathbf{Q}^\dagger = (\mathbf{H}^H\mathbf{H} + \sigma^2\mathbf{I})^{-1}\mathbf{H}^H$. The transmitted bitstream from the $n-$th sender at destination node D_k is extracted with the help of the pseudo-inverse channel matrix $\mathbf{Q}^\dagger$ as follows. The signal is multiplied by the pseudo-inverse:

$$\hat{\mathbf{y}}_{D_k} = \mathbf{Q}^\dagger_{D_k,n}\tilde{\mathbf{y}}_{D_k} = \mathbf{Q}^\dagger_{D_k,n}\mathbf{H}_{D_k,n}\mathbf{x} + \mathbf{Q}^\dagger_{D_k,n}\mathbf{w}_{D_k} \tag{2}$$

where $\mathbf{Q}^\dagger_{D_k,n}$ is a matrix that has all the rows from the pseudo-inverse channel matrix $\mathbf{Q}^\dagger_{D_k}$ minus the $n-$th row, while $\mathbf{H}_{D_k,n}$ is a matrix that has all the columns of the channel matrix $\mathbf{H}_{D_k}$ minus the $n-$th column. In our case we follow an MMSE-OSIC with MRC equalization at the destination node for detection optimality. If we denote with $\bar{\mathbf{y}}$, the ordered version from higher to lower power of the received signals contained in $\tilde{\mathbf{y}}$, then we can apply the OSIC approach along the Y dimension. The destination estimates the highest power $x_n[l]$ symbol (because it is the first in array $\bar{\mathbf{y}}$ is denoted as $\bar{\mathbf{y}}_{D_k,n[1]}$) as

$$\hat{\mathbf{x}}_{D_k,n[l]} = \mathbf{Q}^\dagger_{D_k,n}\bar{\mathbf{y}}_{D_k,n[1]}. \tag{3}$$

IV. Joint 2DSIC and Channel Decoding

In this section we move one step further and we generalize the proposed scheme in the case of coded communications. Low-density parity-check codes (LDPC) are considered for this analysis. Furthermore we employ the sum-product algorithm (SPA) for decoding of the LDPC at each receiver.

One of the contributions of this paper is the joint design of the SPA with the 2DSIC demodulation/detection algorithm we described previously. The rationale of the algorithm is to equalize the symbol and perform SIC, while then use the newly estimated symbols for decoding of the LDPC with the SPA algorithm. The input to the SPA algorithm are the conditional detection probabilities of a symbol. In the proposed scheme, they depend on whether the symbol was detected with X-SIC or Y-SIC. The joint SPA-2DSIC algorithm is described next.

1) Step 1: First the algorithm calculates the conditional detection probabilities for each symbol given that we perform the coherent equalization step as given in (2) and in (3). Without losing generality here we assume BPSK antipodal signaling as before i.e. for the value of the symbol it is $b \in \{\pm 1\}$. Recall that we performed the whitening step for the received signal in (2). That means that the $\hat{\mathbf{y}}_{D_k} - \mathbf{Q}^\dagger_{D_k,n}\mathbf{H}_{D_k,n}\mathbf{x}$ column vector $\sim \mathcal{N}(0, N_0(\sum_{m=1}^{M} g_m^2|h_{R,D_k}|^2 + B))$. Now the probability that the received symbol c_i has the value b given that we observed during the corresponding symbol slot t the signal $\hat{y}_{D_k}$ is:

$$\mathbb{P}(c_i = b|\hat{y}_{D_k}, \mathbf{H}_{D_k,n}) = \frac{1}{\sqrt{\pi N_0}} exp\Big(\frac{-(\hat{y}_{D_k} - \mathbf{Q}^\dagger_{D_k,n}\mathbf{H}_{D_k,n}b)^2}{N_0(\sum_{m=1}^{M} g_m^2|h_{R,D_k}|^2 + B)}\Big)$$

We drop the matrix notation for y above to indicate that it shows the received signal for one symbol position (i in this case). However, note from the above expression that nothing prohibits us from calculating the conditional probabilities for the other symbols that have not yet been detected since the receiver knows $\hat{y}_{D_k}[t], \forall t \in \mathcal{T}$.

2) Step 2: The most important step is the use of the conditional probabilities for X-SIC decoded symbols. When a symbol is decoded with the X-SIC algorithm it is decoded without any impairments caused from other concurrently received symbols because of the direct transmission. Therefore, the confidence for the decoding of the symbol will be higher. We denote this conditional probability as:

$$\mathbb{P}(c_i = b|\hat{y}_{D_k}, h_{S_n,D_k}) = \frac{1}{\sqrt{\pi N_0}} exp(-\frac{(\hat{y}_{D_k} - b)^2}{N_0})$$

Now if

$$\mathbb{P}(c_i = b|\hat{y}_{D_k}, h_{S_n,D_k}) \geq \mathbb{P}(c_i = b|\hat{y}_{D_k}, \mathbf{H}_{D_k,n}),$$

then the i-th variable node in the SPA algorithm changes this value to the newer one since it has a more reliable estimate of a decoded bit value obtained with 2DSIC. Then the SPA runs again and propagates the new estimates as we describe next.

3) Step 3: The SPA algorithm in executed in the next step. The 2DSIC equalization proceeds symbol-by-symbol. For the $i-th$ received codeword bit the variable node i sends to the j-the check node that it is connected the message:

$$q_{i,j} = (\mathbb{P}(c_i = +1|\hat{y}_{D_k}, \mathbf{H}_{D_k,n}), \mathbb{P}(c_i = -1|\hat{y}_{D_k}, \mathbf{H}_{D_k,n}))$$

Since 2DSIC treating sequentially the bits, this means that a check node in the SPA that has not equalized its bits can only provide an initial estimate of $\mathbb{P}(c_i = b|\hat{y}_{D_k}, \mathbf{H}_{D_k,n})$.

Now the check node j after it receives all the messages from the connected variable nodes it sends the following message to variable node i

$$r_{j,i}(0) = \frac{1}{2} + \frac{1}{2}\prod_{i' \neq i}(1 - 2q_{i,j}(1))$$

which also means that $r_{j,i}(1) = 1 - r_{j,i}(0)$.

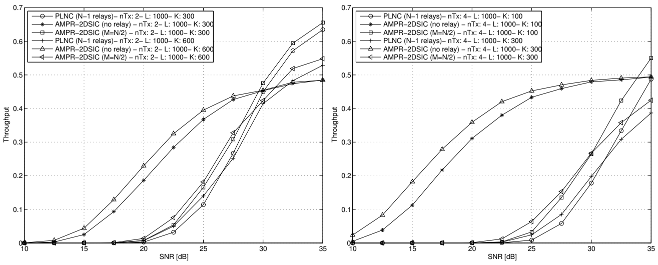
The variable nodes update their response messages according to

$$q_{i,j}(0) = K_{i,j}(1 - P_i)\prod_{j' \neq j} r_{j',i}(0), q_{i,j}(1) = K_{i,j}P_i \prod_{j' \neq j} r_{j',i}(1)$$

At the variable nodes

$$Q_i(0) = K_i(1 - P_i)\prod_{j'} r_{j',i}(0), Q_i(1) = K_i P_i \prod_{j'} r_{j',i}(1)$$

So the decoder outputs a 1 as its estimate for bit i if $Q_i(1) > Q_i(0)$ or 0 otherwise.



(a) (b)

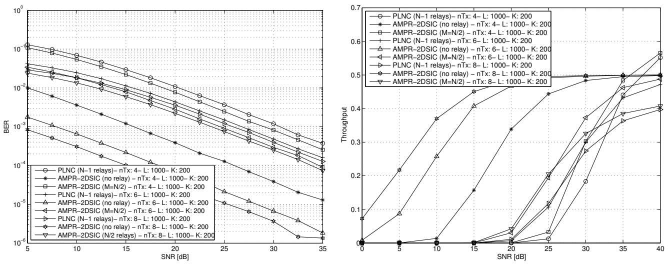
Fig. 4. Results for two and four users and different K.

V. Performance Evaluation

Different systems were evaluated in terms of BER and throughput under different channel conditions through Monte Carlo simulations. AMPR-2DSIC with no relay is evaluated first. Also AMPR-2DSIC is tested with a single relay and we assume that an equal number of broadcast and forwarding phases occurs, i.e. $B = N/2$ and $M = N/2$ respectively. Also PLNC is tested when $N-1$ mixed signals are forwarded from the relays [6]. Packets were generated with lengths taken from a uniform distribution so that their average length is L. The results are averages of $1{,}000$ packet transmissions. The channel bandwidth is 20 MHz, AWGN with noise variance 10^{-9} W/Hz at every node. We also used a Rayleigh fading wireless channel model. The channel transfer functions between the nodes vary independently for each transmitted packet but they are characterized by the same average SNR. CSI knowledge is assumed at the receiver. Furthermore, in order to be fair, we assumed that the transmission rate from the sender to the relay and from the relay to the destination is double from the direct mode [8]. LPDC code of 32Kbits was employed.

In Fig. 4(a) we see the performance in terms of throughput for different average number of free symbol slots K and two users. As K is increased the average degree of overlap is decreased between different packets. For PLNC and AMPR-2DSIC ($M = N/2$) this has a positive impact on the BER that is reduced since the Y-Dimension SIC and decoding algorithms have to decode fewer superimposed symbols. However, the BER reduction comes at the cost of lower utilization of the available symbol slots that remain ”empty”. It appears that this has the highest impact on the final throughput that drops in the high SNR regime as K is increased to 600.

Another important result is the different behavior of AMPR-2DSIC (no relay) that presents generally better performance in the low SNR regime. This is primarily due to lack of noise amplification that occurs at the relays with the other protocols. Also AMPR-2DSIC (no relay) performs better as fewer symbols are superimposed since this helps with the decoding process. Therefore, for two users in the low SNR regime it is more efficient to minimize the level of overlap while for increasing channel SNR the level of packet overlap should be maximized. A final interesting result in Fig. 4(a) is that as the SNR is increased the performance gap between



(a) BER. (b) Throughput.

Fig. 5. Results for different number of users N and $K = 200$ for AMPR-2DSIC and PLNC.

PLNC and AMPR-2DSIC is decreased. The reason for this is the reduced impact of noise on the relays in this SNR regime, which makes PLNC that uses several relays also efficient. The performance gap is also observed for the lower value of $K = 300$, where significant overlap of symbols occurs.

Next we consider four concurrent users. It is important to see in Fig 4(b) that AMPR-2DSIC (no relay) maintains the same performance. The results are becoming worse for the other protocols when four nodes broadcast at the same time while the performance drop is faster for an increasing K as Fig 4(b) indicates. However, the protocol can still provide benefit when $K \leq 300$ and for the high SNR regime when compared to a AMPR-2DSIC (no relay). Also the performance gap between the two protocols in the high SNR regime is higher in this case when compared to the previous experiment of $N = 2$. This occurs precisely because more nodes superimpose their symbols and the number of detected symbols by 2DSIC is higher.

We now examine a constant $K = 200$ for all the tested protocols and different number of users N in Fig. 5(b). A first observation we can make from the throughput results in Fig. 5(b) is that there is a minor decrease in the performance difference between AMPR-2DSIC ($M = N/2$) over PLNC for lower values of N. However, the difference becomes less important when N is increased. The reason is again that for higher SNR lower noise amplification occurs. The most interesting results can be observed in the low SNR regime. In this regime the performance differences between AMPR-2DSIC (no relay) and the other protocols is attributed to the severe noise amplification that hurts the other protocols that use the relays. Also it is very crucial to see that AMPR-2DSIC (no relay) performs better as the number of nodes is increased even with this level of asynchrony. Of course the BER that all protocols can achieve in Fig. 5(a) is higher for fewer transmitting nodes contrary to the results for AMPR-2DSIC (no relay) where BER is reduced for more nodes. This is because of the time diversity benefit that is exploited by this protocol. The BER of both protocols that use relays converges more in the high SNR regime leading thus again to the superiority over AMPR-2DSIC (no relay). However, for the systems that use relays the optimal choice in the high SNR regime is $N = 2$ users as shown in Fig. 4(a).

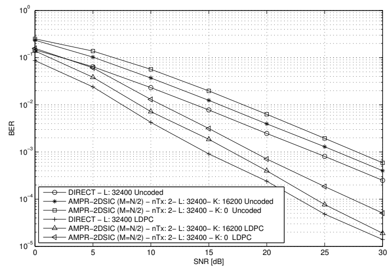


Fig. 6. Results for joint AMPR and LDPC decoding.

The BER performance of the joint 2DSIC and SPA algorithms for decoding of the LDPC is presented in Fig. 6 for the case of direct transmission, AMPR-2DSIC with full overlap and AMPR-2DSIC with 50% packet overlap. We see the significant performance benefits of the scheme where coded communication is employed over AMPR-2DSIC without coding. It is important to stress the fact that the performance of the proposed scheme is independent of the coding gains.

VI. Conclusions

In this paper we presented a new SIC algorithm that exploits asynchronous over-the-air superimposed transmissions for MPR. The goal of the proposed scheme is to be able to increase the MPR performance but at the same time remove impractical restrictions to the higher layer protocols like scheduling of packets for perfect alignment of have the same length. Performance results demonstrate that in the low SNR regime no relay is needed and the use of more concurrent users is beneficial regardless of the level of asynchrony. For better channel quality, asynchrony has higher impact on performance which means that the number of concurrent users should be small while the use of relays is more beneficial. The highest benefits are observed when the proposed AMPR decoding algorithm is jointly designed with the channel decoder.

References

[1] L. Tong, Q. Zhao, and G. Mergen, "Multipacket reception in random access wireless networks: from signal processing to optimal medium access control," *Communications Magazine, IEEE*, vol. 39, no. 11, pp. 108 –112, nov 2001.

[2] M. Tsatsanis, R. Zhang, and S. Banerjee, "Network-assisted diversity for random access wireless networks," *Signal Processing, IEEE Transactions on*, vol. 48, no. 3, pp. 702 –711, mar 2000.

[3] M. Dankberg, M. Miller, and M. Mulligan, "Self-interference cancellation for two-party relayed communication," United States Patent 5596439, January 1997.

[4] S. Zhang, S. C. Liew, and P. P. Lam, "Hot topic: Physical-layer network coding," in *MobiCom*, 2006.

[5] S. Katti, S. Gollakota, and D. Katabi, "Embracing Wireless Interference: Analog Network Coding," in *SIGCOMM*, 2007.

[6] A. Argyriou and A. Pandharipande, "Cooperative protocol for analog network coding in distributed wireless networks," *IEEE Transactions on Wireless Communications*, vol. 8, no. 10, pp. 2014–2023, October 2010.

[7] A. Goldsmith, *Wireless Communications*. Cambridge University Press, 2005.

[8] J. N. Laneman, D. Tse, and G. Wornell, "Cooperative diversity in wireless networks: Efficient protocols and outage behavior," *IEEE Transactions on Information Theory*, vol. 50, no. 12, pp. 3062–3080, December 2004.