SigSag: Iterative Detection through Soft Message-Passing

Arash Saber Tehrani , Alexandros G. Dimakis , Michael J. Neely

***Abstract*—The multiple-access framework of ZigZag decoding [1] is a useful technique for combating interference via multiple repeated transmissions, and is known to be compatible with distributed random access protocols. However, in the presence of noise this type of decoding can magnify errors, particularly when packet sizes are large. We present a simple soft-decoding version, called SigSag, that improves performance. We show that for two users, collisions result in a cycle-free factor graph that can be optimally decoded via belief propagation. For collisions between more than two users, we show that if a simple bit-permutation is used then the graph is locally tree-like with high probability, and hence belief propagation is near optimal. Through simulations we show that our scheme performs better than coordinated collision-free time division multiple access (TDMA) and the ZigZag decoder.**

I. Introduction

Despite the substantial amount of theoretical work in multiuser detection and inference cancellation, most implementations rely on carrier sense (CSMA) to limit collisions while CDMA receivers decode each user by treating interference as noise. One important step towards practical systems that decode interfering users was ZigZag decoding by Gollakota and Katabi [1] that relies on the 802.11 MAC. This restricts the design space into a very simple repetition of packets when there are collisions and decoding failures. We build on the same assumptions as the original ZigZag framework. However, we develop a soft-decoding version, called SigSag, that is just as simple to implement but results in significantly improved reception rates.

Specifically, we assume there are N users, each having a packet of B bits, trying to communicate with an access point (AP) (See Fig. 1). Each user relies on carrier sensing to detect if other users are transmitting. If this fails (the case of a *hidden terminal*) there is interference at the AP, modeled by a simple linear superposition of the symbols plus noise. In this paper we only consider the worst case, where carrier sensing constantly fails and there are always packet collisions. The improved decoding probabilities we obtain in this paper can be used to improve performance of higher layer multi-access protocols that incorporate ZigZag-like collision frames, such as those in [23]. Under the 802.11 protocol, each sender retransmits its packet until it receives an acknowledgment that the AP successfully decoded it. In our model, each of the N users transmits its packet N times and the AP receives linear equations involving the sum of the collided symbols plus noise. Similarly to prior work, we model the random

The authors are with the Electrical Engineering department at the University of Southern California, Los Angles, CA.

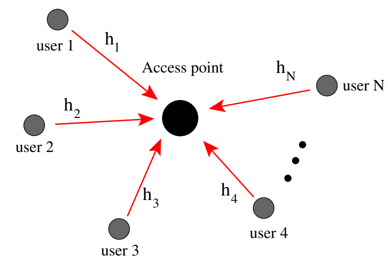

Fig. 1. All nodes send data to the access point (AP) together.

delay offsets (jitter) of the 802.11 protocol as follows: as packets from two or more users collide, each user chooses a random bit-offset uniform over $[0, W]$ and transmits its packet again. The maximum likelihood (ML) detection (also referred to as a multiuser demapping) problem consists of finding the most likely user symbols given the noise statistics. In the high-SNR case (when the noise is negligible) this simply reduces to solving linear equations, while for the noisy case it becomes a statistical inference problem which is well known to be computationally intractable.

Our contributions: We show that the ZigZag algorithm can be seen as an instance of belief propagation in the high-SNR limit, essentially attempting to solve linear equations by back-substitution only. We introduce a new algorithm called SigSag that *exploits* and *maintains* soft information about each symbol. If ZigZag is seen as hard-decision belief propagation, our algorithm is a natural generalization that maintains likelihoods and runs in a loopy manner on the factor graph created by the linear equations formed by collided packets.

We show that for $N = 2$ users, our iterative soft message-passing algorithm is optimal, by establishing that the corresponding factor graph is cycle-free with high probability. Further, for $N \geq 3$ users, we show that the factor graph is tree-like by establishing that the local neighborhood is cycle-free with high probability. Our results establish that while the maximum-likelihood inference problem is NP-hard in general, the random jitter bit-offsets and bit permutations in packets create easy instances with high probability. For these instances the maximum likelihood solution is found by our soft message-passing detector. The last part of our theoretical analysis shows that random jitter and fading can critically influence performance even in the high-SNR regime. Our preliminary experimental analysis shows the substantial practical benefits of using SigSag soft message-passing compared to ZigZag and Time Division Multiple Access (TDMA), also often referred to as collision free scheduling.

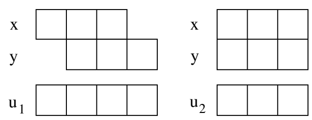

Fig. 2. Two consecutive collisions of two packets $\mathbf{x}$ and $\mathbf{y}$ of $B = 3$ symbols sent by $N = 2$ users. Both packets are transmitted twice and the AP receives $\overline{u}_1$ and $\overline{u}_2$.

A. System Model

As shown in Fig. 1, we have N users trying to communicate to an access point (AP). Each user re-transmits N times and the packets collide forming linear equations at the AP. We assume that the users can permute the bits of their packets before each transmission, as discussed in [11]. We further assume a block fading channel model where data sent by user i on the c^{th} transmission are attenuated by coefficients $h_i^{(c)}$ assumed known at the receiver[1]. Under this model, the AP receives the signal

$$\overline{u}_c = u_c + \nu^{(c)} = \sum_{i=1}^{N} h_i^{(c)} \mathbf{T}_{w(i,c)}(x_i) + \nu^{(c)}, \qquad (1)$$

where $c \in \{1, 2, \ldots, N\}$ is the collision round, $x_i = [x_{i,1}, x_{i,2}, \ldots, x_{i,B}]$ is the packet (assuming BPSK, $x_{i,j} = \pm 1$) sent by the i^{th} user, $\nu^{(c)}$ is the channel noise vector of c^{th} collision, assumed to be independent and identically distributed Gaussian for simplicity. Finally, $\mathbf{T}_{w(i,c)}(.)$ represents the jitter or time delay user i chooses randomly before the c^{th} transmission. Note that the above does not mean that the b^{th} symbol $b \in \{1, \ldots, B\}$ of the packets x_i, $i \in \{1, \ldots, N\}$ combine with each other. The notation is only to keep the exposition clear. As a result the whole system can be modeled as $A\mathbf{x} + \nu = \overline{\mathbf{u}}$ or $A\mathbf{x} = \mathbf{u}$ where A is the *collision matrix*, $\mathbf{x} = [x_1, x_2, \ldots, x_N]^T$, and $\mathbf{u} = [u_1, u_2, \ldots, u_N]^T$.

As an example, consider the case shown in Fig. 2 which depicts two consecutive collisions of the packets sent by $N = 2$ users. Notice that each collision results in at least B linear equations. As explained, the whole system of linear equations can be rephrased in the form of $A\mathbf{x} = \mathbf{u}$, where here $\mathbf{x} = [\mathbf{x}, \mathbf{y}]^2$ and $u = [u_1, u_2]^T$. The set of linear equations corresponding to the collision patterns shown in Fig. 2 are

$$\begin{aligned}
\overline{u}_{1,1} &= h_1^{(1)}\mathbf{x}_1 + \nu_1^{(1)}\\
\overline{u}_{1,2} &= h_1^{(1)}\mathbf{x}_2 + h_2^{(1)}\mathbf{y}_1 + \nu_2^{(1)}\\
\overline{u}_{1,3} &= h_1^{(1)}\mathbf{x}_3 + h_2^{(1)}\mathbf{y}_2 + \nu_3^{(1)}\\
\overline{u}_{1,4} &= h_2^{(1)}\mathbf{y}_3 + \nu_4^{(1)}\\
\overline{u}_{2,1} &= h_1^{(2)}\mathbf{x}_1 + h_2^{(2)}\mathbf{y}_1 + \nu_1^{(2)}\\
\overline{u}_{2,2} &= h_1^{(2)}\mathbf{x}_2 + h_2^{(2)}\mathbf{y}_2 + \nu_2^{(2)}\\
\overline{u}_{2,3} &= h_1^{(2)}\mathbf{x}_3 + h_2^{(2)}\mathbf{y}_3 + \nu_3^{(2)}
\end{aligned}$$

[1]Note that for estimating the fading coefficients $h_i^{(c)}$ at the receiver, the 802.11 adds a known preamble sequence to each packet. Further, for this estimation to be exact, we require fading to be extremely slow.

[2]For more clarity, here we call the two packets by $\mathbf{x}$ and $\mathbf{y}$ instead of x_1 and x_2.

and the corresponding collision matrix A is

$$A = \begin{bmatrix}
h_1^{(1)} & & & & & \\
& h_1^{(1)} & & h_2^{(1)} & & \\
& & h_1^{(1)} & & h_2^{(1)} & \\
& & & & & h_2^{(1)}\\
h_1^{(2)} & & & h_2^{(2)} & & \\
& h_1^{(2)} & & & h_2^{(2)} & \\
& & h_1^{(2)} & & & h_2^{(2)}
\end{bmatrix}.$$

Notice that if there was no noise (all $v_i^{(c)} = 0$), the optimal decoder would simply have to solve these linear equations. In the presence of noise, optimal decoding would correspond to finding which vectors $\mathbf{x}_i \in \{\pm 1\}^B$, $\mathbf{y}_i \in \{\pm 1\}^B$ have the highest likelihood under the noise statistics, which reduces to a computationally intractable integer least-square problem.

II. Related Work

The amount of related work in communications and information theory on multiuser joint detection and decoding is overwhelming and without trying to be comprehensive we will point the interested reader to [2], [4], [8]–[10], [12], and references therein. A comprehensive overview on the recent advances in interference cancellation can be found in [14].

ZigZag decoding [1] is the closest to our work. The approach taken in this paper is to restrict the design space to packet repetitions and hence maintain compatibility with 802.11. One important advantage of this approach is that, unlike *joint decoding* methods, senders do not need to reduce their rate in order for the AP to be able to decode the collisions.

By exploiting the small variations in the transmission times, ZigZag attempts to decode the packets of each user by solving the linear equations $A\mathbf{x} = \mathbf{u}$ by back-substitution. This version of ZigZag (the Forward ZigZag) is identical to the belief propagation decoder for the BEC (see e.g. [5], [7]), also known as the leaf-stripping decoder which looks for a degree one variable, makes a hard decision (decides either ± 1) about it and removes it from the factor graph. This algorithm is feasible because the random jitter allows some symbols to appear without interference (i.e. have degree 1) with high probability. An improved version of ZigZag runs this algorithm from different starting points (Forward-Backward [1]) and then combines partial soft-information from the two executions to decode the variables. Our algorithm is essentially an optimized version of this idea that keeps track of how this soft information evolves and deferring the rounding decisions until the end. Remap [11] is an extension of ZigZag where senders permute their data after the first transmission and Chorus [13] resolves collisions over a *multi-hop* network to improve latency and transmission diversity of the network.

Our work is closely related to wireless network coding (see e.g. [16], [18], [21]) and cross-layer designs that exploit coding [17]. In our case, it is the interference that creates the code, similarly to [20]. Our technical analysis relies on iterative message passing algorithms and the theory developed for sparse-graph codes [5], [7], [15], [19].

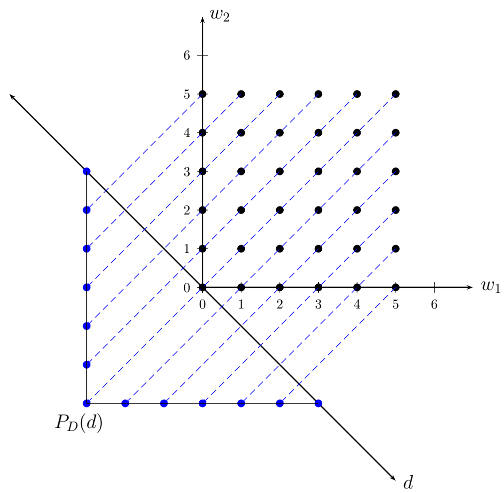

Fig. 3. The probability mass function of $d = w_1 - w_2$ when w_i's are discrete random variables uniformly chosen from $\{0, 1, \ldots, W\}$ (here $W = 5$).

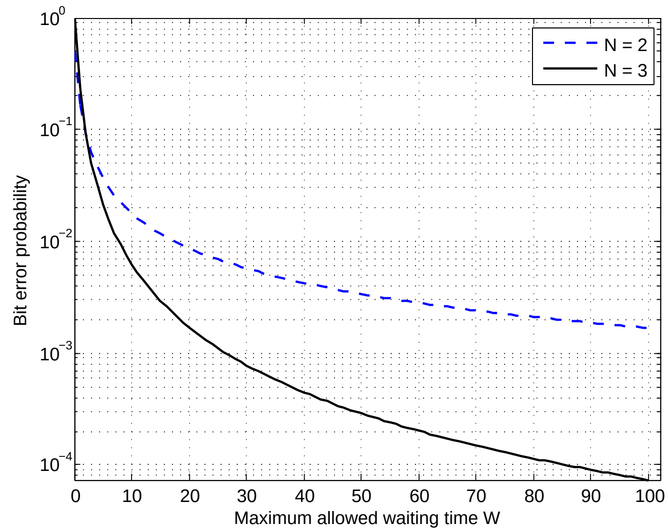

Fig. 4. The BER threshold of the ZigZag decoder for $N = 2$ and $N = 3$ user.

A. Shortcomings of ZigZag

In this paper we extend ZigZag decoding [1], addressing two main drawbacks:

1) ZigZag can fail to decode when back-substitution is not possible.
2) Noise is accumulated as the ZigZag decoder advances through the packet.

1) Back-substitution Failure: Here, we focus on the first drawback of ZigZag and show that it causes a threshold on the bit error probability that ZigZag can achieve. This threshold depends on the number of collided packets N and the maximum waiting time before each transmission W, but not on the SNR.

Consider the case of $N = 2$ users where each user waits for a random time uniformly chosen from $\{0, 1, \ldots, W\}$ before each transmission. Assuming the users have similar SNR's[3], the zigzag decoder fails to retrieve the packets if the two collision patterns are identical. In other words, if we define $w_i^{(c)}$ for $i \in \{1, 2\}$ to be the waiting time of user i before its c^{th} transmission, then the patterns will be identical if $d^{(1)} = w_1^{(1)} - w_2^{(1)}$ is equal to $d^{(2)} = w_1^{(2)} - w_2^{(2)}$, i.e. the waiting time difference of two users for both transmissions are equal.[4]

As shown in Fig. 3 the probability mass function of $d = w_1 - w_2$ where w_1 and w_2 are uniformly distributed over $\{0, \ldots, W\}$ is

$$p_D(d = k) \quad = \begin{cases} \frac{(W+1)-|k|}{(W+1)^2} & \text{if } |k| \leq W \\ 0 & \text{otherwise} \end{cases}$$

and the probability of having identical collision patterns in both transmissions is

$$\sum_{k=-W}^{W} (P_D(d = k))^2 = \frac{1}{3} \frac{2W^2 + 4W + 3}{(W + 1)^3}.$$

When the above happens, the ZigZag is unable to decode the packets since it cannot initiate the back-substitution.

For $N = 3$ users, the error happens if either two out of three collision patterns are identical, or two packets collide in the same way in all three collisions. As we are only looking for a lower bound, we consider only the latter case and we have

$$P\,[\text{failure}] > \textstyle\sum_{k=-W}^{W} (P_D(d = k))^3$$

where $P_D(d = k)^3$ is the probability that users 1 and 2 have a delay difference of k on all 3 collision rounds (which is the same probability if computed between users 1 and 3 or users 2 and 3).

Fig. 4 shows the bound on the error probability of the ZigZag decoder for $N = 2, 3$ users as a function of the maximum waiting time W. Again note that the above bound holds, regardless of the SNR.

2) Error Accumulation: The second shortcoming of ZigZag is error accumulation which can be described as follows. ZigZag decoding relies on continuously repeating two steps: i) finding free chunks and decoding them, and ii) removing the decoded bits from other collisions to produce new free chunks (back-substitution). At each decoding step, ZigZag makes a hard decision on the value of the bit which causes ZigZag to aggregate noise as it decodes through the packet. This aggregation becomes more severe as the length of the collided packets grow and worsens the performance of ZigZag. Fig. 5 shows the bit error probability of ZigZag for $N = 2$ users at $SNR = 5dB$, as the length of the packets B grows.

On the other hand, as it is shown in Fig. 6, the performance of our proposed SigSag algorithm improves as the length of the packets grow. This is due to the fact that the factor graph on which we run the message passing for SigSag becomes more

[3]If the SNR's are too different, ZigZag can decode the packets using the capture effect.

[4]Note that a collision (the bit-offset d) can be detected at the receiver by adding a known preamble to each packet, as explained in [1]

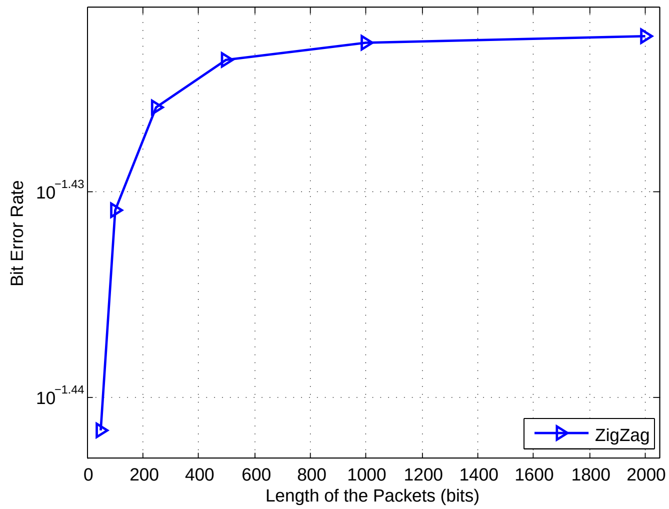

Fig. 5. ZigZag error accumulation for $N = 2$ users at $SNR = 5dB$.

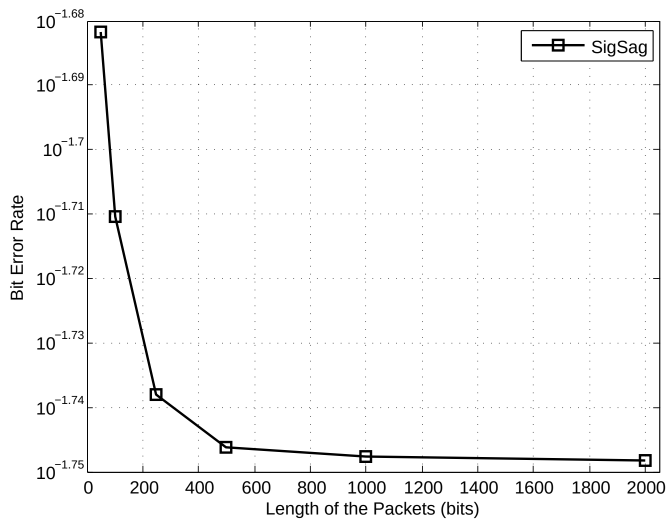

Fig. 6. Performance of SigSag as the length of the packets grow for $N = 2$ users at $SNR = 5dB$.

tree-like as the length of the packets grow. This is proved in section IV-B.

III. SigSag Algorithm

In this section we introduce SigSag, our soft iterative message-passing decoder, which attempts to find a good approximation decoding for the users packets. There are two variations of SigSag with slightly different objectives: The first uses iterative sum-product to compute the marginals for each bit and hence minimize the bit-error rate. The second uses max-product to compute the jointly most likely configuration, attempting to minimize the block-error rate. When executed on factor graphs with cycles these algorithms can be suboptimal and in practice we observe that max-product has slightly better performance for the cases tested.

We derive the messages for the first few steps of the max-product algorithm for the check f_{11}, as explained in [6], [7], on the factor graph shown in Fig. 7 which corresponds to the

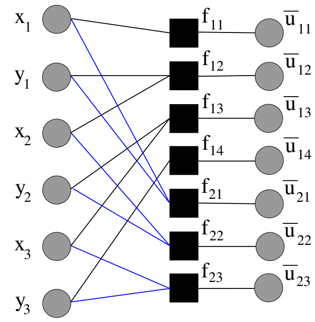

Fig. 7. Two consecutive collisions of two senders and the factor graph representation of the pattern.

collision pattern shown in Fig. 2. Here, $\mu_{a \longrightarrow b}(x)$ represents the probability distribution function of x estimated at node a and sent as a message to node b. This distribution (sent over link $a \longrightarrow b$) is estimated from the previous messages that were sent on all incoming links to node a (other than the link connecting it to b). The vector $\overline{u}_c$ is the noisy observation at the receiver on the c^{th} collision. Also let u_c be the received signal if the channel was noise-free, i.e. $\overline{u}_c = u_c + \nu$ where ν is the noise vector.

step 1:
$$\mu_{\overline{u}_{11} \longrightarrow f_{11}}(u_{11}) = \begin{cases} p\{u_{11} = h_1^{(1)} | \overline{u}_{11}\} \\ p\{u_{11} = -h_1^{(1)} | \overline{u}_{11}\} \end{cases}$$

step 2:
$$\mu_{f_{11} \longrightarrow x_1}(x_1) = \begin{cases} p\{x_1 = 1\} = p\{u_{11} = h_1^{(1)} | \overline{u}_{11}\} \\ p\{x_1 = -1\} = p\{u_{11} = -h_1^{(1)} | \overline{u}_{11}\} \end{cases}$$

step 3:
$$\mu_{x_1 \longrightarrow f_{21}}(x_1) = \begin{cases} p\{x_1 = 1\} \\ p\{x_1 = -1\} \end{cases}$$

Step 4 is quite similar to step 1, only the degree of the nodes are different:
$$\mu_{\overline{u}_{21} \longrightarrow f_{21}}(u_{21}) = \begin{cases} p\{u_{21} = h_1^{(2)} + h_2^{(2)} | \overline{u}_{21}\} \\ p\{u_{21} = h_1^{(2)} - h_2^{(2)} | \overline{u}_{21}\} \\ p\{u_{21} = -h_1^{(2)} + h_2^{(2)} | \overline{u}_{21}\} \\ p\{u_{21} = -h_1^{(2)} - h_2^{(2)} | \overline{u}_{21}\} \end{cases}$$

In step 5, the algorithm applies the max operation to find the most probable configuration for each mass point $\{1, -1\}$:
$$\mu_{f_{21} \rightarrow y_1}(y_1) = \begin{cases} p\{y_1 = 1\} = \\ \frac{1}{z} \max \Big(p\{u_{21} = h_1^{(2)} + h_2^{(2)} | \overline{u}_{21}\} p\{x_1 = 1\}, \\ \qquad p\{u_{21} = -h_1^{(2)} + h_2^{(2)} | \overline{u}_{21}\} p\{x_1 = -1\} \Big) \\ p\{y_1 = -1\} = \\ \frac{1}{z} \max \Big(p\{u_{21} = h_1^{(2)} - h_2^{(2)} | \overline{u}_{21}\} p\{x_1 = 1\}, \\ \qquad p\{u_{21} = -h_1^{(2)} - h_2^{(2)} | \overline{u}_{21}\} p\{x_1 = -1\} \Big) \end{cases}$$

where z is a normalization factor. By changing the *max* operation to *sum* the algorithm becomes the *sum-product*

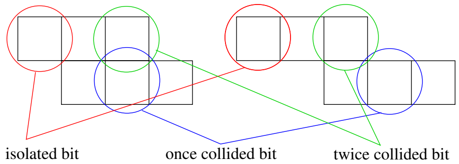

Fig. 8. Demonstration of isolated, once collided and twice collided bits for the two user scenario.

algorithm. Further, instead of propagating the probabilities, one can transmit log-likelihood ratios

$$m_{f\longrightarrow x}(x) \quad = \log\frac{p\{x=1\}}{p\{x=-1\}}$$
$$m_{x\longrightarrow f}(x) \quad = \log\frac{p\{x=1\}}{p\{x=-1\}}$$

for factor node f and variable node x, where $p\{x=1\}$, and $p\{x=-1\}$ are derived as mentioned earlier. For instance, for the sum-product algorithm, $m_{f_{21}\longrightarrow y_1}(y_1)$ from the above example can be computed to be

$$\begin{aligned} m_{f_{21}\longrightarrow y_1} \quad (y_1) &= \log\frac{p\{y_1=1\}}{p\{y_1=-1\}} \\ &= \log\frac{p\{u_{21}=h_1^{(2)}+h_2^{(2)}|\overline{u}_{21}\}e^{m_{x_1\longrightarrow f_{21}}(x_1)}}{p\{u_{21}=h_1^{(2)}-h_2^{(2)}|\overline{u}_{21}\}e^{m_{x_1\longrightarrow f_{21}}(x_1)}} \\ &\qquad \frac{\cdots+p\{u_{21}=-h_1^{(2)}+h_2^{(2)}|\overline{u}_{21}\}}{\cdots+p\{u_{21}=-h_1^{(2)}-h_2^{(2)}|\overline{u}_{21}\}} \end{aligned}$$

IV. Theoretical Analysis

In this section, we analyze the performance of SigSag theoretically, and prove that it is ML for two users and close to ML for more than two users. Then we consider the high SNR regime and discuss the importance of jitter and fading.

A. Factor graph is cycle-free for two users

We show that for the case of two users the factor graph of the received signal is cycle-free if the bit-offsets of the two consecutive collisions are different. This result does not require permutations of bits with every retransmission. We start by observing that there are three different kinds of bits

1) Isolated or degree zero bits: are received in free symbols, i.e. without colliding with any other bit in both collision rounds.
2) Once collided or degree one bits: collide with a bit from the other packet in one collision and received unharmed on the next collision.
3) Twice collided or degree two bits: collide with different bits of the other packet in each collision.

The above node classes are shown in Fig. 8.

Since for two users every function node has degree at most two, we can remove the factor nodes and consider two bits connected if they collide with each other.

Theorem 1: In the case of two users transmitting without permuting their bits, the resulting factor graph is cycle-free if the bit-offsets of the two consecutive collisions are different.

Proof: First, observe that there can be no cycle of length two (including factor nodes) since a bit cannot collide with itself, and there cannot be any cycle of length four since it means the bit-offset differences are identical. Therefore, we are only concerned about cycles of higher length. Here we use the same notation introduced in II-A1, where $d^{(1)}$ and $d^{(2)}$ are the time difference between user transmissions in first and second collisions. Arrange the variable nodes according to their order of appearance in the packets. In other words, arrange them in two columns $x_1, x_2, \ldots, x_B$ and $y_1, y_2, \ldots, y_B$. Beginning from an arbitrary variable node, we try to draw a cycle (length more than four) by connecting the variable nodes through arbitrary factor nodes. Assume that we intend to make a cycle which begins and ends at x_i where x_i is a degree two node. First we connect x_i to y_j, and then we connect y_j to x_k. Note that these two edges have already determined the pattern of the both collisions, since $d^{(1)} = i-j$ and $d^{(2)} = k-j$. For example, we can deduce that the node x_i collides with y_j on the first collision and with $y_{i-d^{(2)}} = y_{i-k+j}$ on the second one. We consider two cases, either $k > i$ or $k < i$ (Note that $k = i$ contradicts the assumption since $d^{(1)} = d^{(2)}$). Consider the former $k > i$. As mentioned before, We want to continue connecting the bits from y_j and get back to x_i. The pass is $\{x_i, y_j, x_k, y_{k-d^{(1)}}, x_{k-d^{(1)}+d^{(2)}}, \ldots\}$. We see that on every return to an "x" bit, the index increases by exactly $k-i$ (note that $(k-d^{(1)}+d^{(2)})-k = k-i$). Thus the chain must increase away from x_i until it reaches a degree-1 node that terminates and shows this is not a cycle. The same argument holds for the case $k < i$, only this time the chain moves toward the start of the packets and terminates there. ■

We proved that if the bit-offsets of the the two collisions are different the factor graph is cycle-free. In this case, it is trivial to check that each isolated bit form a path together with its neighboring factor nodes while all degree one and degree two bits form a path starting and ending in a degree one bit which has degree two bits in the middle. Since message passing on cycle-free factor graphs is optimal [6], our soft receiver is performing ML detection for two users.

B. Tree-like factor graph for multiple users

We now consider any number of users $N \geq 2$, but assume that the users permute the bits of the packets randomly before each transmission. Here, it is shown that the factor graph formed from the collision pattern is almost tree-like as the length of the packets B grows, and thus, the belief propagation result is close to ML. In the following, $\mathbf{N}_e^{\ell}$ denotes the directed neighborhood of depth ℓ of $e = (v, c)$ which is defined as the induced subgraph containing all edges and nodes on paths $e_1, \ldots, e_\ell$ starting from v such that $e_1 \neq v$.

Theorem 2: The left-regular bipartite factor graph G resulted form collision of N packets of length B sent by N users with symbol permutation is locally tree-like with high probability. Specifically,

$$\Pr\{\mathbf{N}_e^{2\ell^*} \text{ is not tree like}\} \leq \frac{s}{B}$$

where $e = (v, c)$ is a randomly chosen edge of the graph, 2ℓ is a fixed depth, and s is a suitable constant that depends on ℓ, N, but not on B.

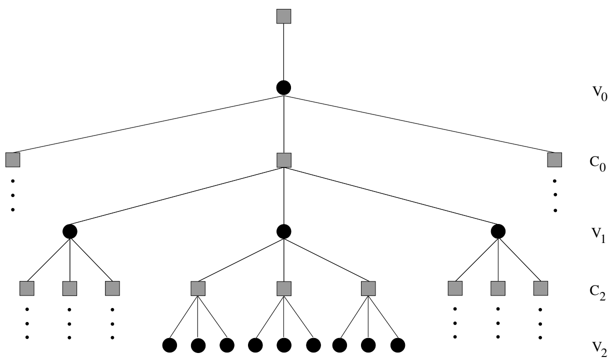

Fig. 9. Section of the the directed neighborhood of depth 3 of an edge.

Proof: This proof is using a similar technique as the local tree-like proofs for random sparse graphs developed for Low Density Parity Check (LDPC) codes [7]. The ensemble of random graphs we are dealing with, formed by the random jitter and permutations, are different from the LDPC ensembles but still the same inductive approach works. Note that all the variable nodes v have degree $d_v = N$ and the degree of the check nodes d_c is less than or equal to N ($d_c \leq N$). In other words, each bit appears in exactly N equations and each equation contains at most N variables. Since increasing the degree of the checks only increase the probability of existence of cycles, without loss of generality we assume $d_c = N$ as shown in Fig. 9. At each level i there are $V_i = (d_v-1)^i(d_c-1)^i = (N-1)^{2i}$ variable nodes and $C_i = (d_v-1)^{i+1}(d_c-1)^i = (N-1)^{2i+1}$ check nodes. As a result, assuming that $\mathbf{N}_e^{2\ell}$ is tree-like, there are

$$\overline{V}_\ell = \sum_{i=0}^{\ell} V_i$$

$$\begin{aligned}\overline{V}_\ell &= \sum_{i=0}^{\ell}(d_v-1)^i(d_c-1)^i \\ &= \sum_{i=0}^{\ell}(N-1)^{2i} \\ &= \frac{(N-1)^{2\ell+2}-1}{(N-1)^2-1} = \frac{(N-1)^{2\ell+2}-1}{N(N-2)}\end{aligned}$$

variable nodes and

$$\begin{aligned}\overline{C}_\ell &= 1 + \sum_{i=0}^{\ell-1} C_i \\ &= 1 + (d_v-1)\sum_{i=0}^{\ell-1}(d_v-1)^i(d_c-1)^i \\ &= 1 + (N-1)\sum_{i=0}^{\ell-1}(N-1)^{2i} \\ &= 1 + (N-1)\frac{(N-1)^{2\ell}-1}{(N-1)^2-1}\end{aligned}$$

check nodes in it. Recall that the collision matrix A is $M \times NB$, where $NB \leq M \leq N(B+W)$. We want to compute the probability that the $\mathbf{N}_e^{2\ell+1}$ is tree-like. For that, we consider a variable node in the $2\ell^{th}$ depth. The probability that its k^{th} edge does not create a loop assuming that its previous $k-1$ edges have not, is

$$P \geq \frac{(M-\overline{C}_\ell-k)d_c}{Md_c-\overline{C}_\ell-k} \quad \geq 1-\tfrac{\overline{C}_{\ell^*}}{M},$$

for $\ell \leq \ell^*$. Thus the probability that $\mathbf{N}_e^{2\ell+1}$ is tree-like given that $\mathbf{N}_e^{2\ell}$ is tree-like is lower bounded by $(1-\frac{\overline{C}_{\ell^*}}{M})^{C_{\ell+1}}$. Notice that each check is allowed to have at most one neighbor from each packet. So consider the variable node at level zero (V_0) to belong to the i^{th} packet. We use R_ℓ for number of bits from the i^{th} packet at level ℓ, and (since number of bits from each of other packets are equal at each level) K_ℓ for number of bits from each one of the other packets at level ℓ. Apparently, at level $\ell = 0$, we have $R_0 = 1$, and $K_0 = 0$. It is easy to check that $R_\ell = (N-1)K_{\ell-1}$ and $K_\ell = \frac{V_\ell - R_\ell}{N-1}$ for all ℓ. Unlike the previous part, the number of bits at each level are different for different users. However, we can bound the total number of bits from each user at level ℓ by $c\frac{\overline{V}_\ell}{N}$, i.e $c\frac{\overline{V}_\ell}{N} \geq \sum_{\ell=0}^{\ell^*} R_\ell$ and $c\frac{\overline{V}_\ell}{N} \geq \sum_{\ell=0}^{\ell^*} K_\ell$ where c is a positive scalar (for example $c = 2$). As a result, the same as above, we can show that the outgoing edges of the check nodes at depth $2\ell+1$ will not make a loop with probability

$$Q \geq \tfrac{(B-c\frac{\overline{V}_\ell}{N})d_v}{Bd_v - \frac{\overline{V}_\ell}{N}} \quad \geq 1 - c\frac{\overline{V}_{\ell^*}}{NB}.$$

So, the probability that $\mathbf{N}_e^{2\ell+2}$ is tree-like given that $\mathbf{N}_e^{2\ell+1}$ is tree like is lower-bounded by $(1-c\frac{\overline{V}_{\ell^*}}{NB})^{V_{\ell+1}}$. Thus,

$$\begin{aligned}\Pr\{\mathbf{N}_e^{2\ell} \text{ is tree like}\} &\geq (1-c\frac{\overline{V}_{\ell^*}}{NB})^{\overline{V}_{\ell^*}}(1-\frac{\overline{C}_{\ell^*}}{M})^{\overline{C}_{\ell^*}} \\ &\geq (1-c\frac{\overline{V}_{\ell^*}}{NB})^{\overline{V}_{\ell^*}}(1-\frac{\overline{C}_{\ell^*}}{N(B+W)})^{\overline{C}_{\ell^*}}.\end{aligned}$$

For large enough n,

$$\begin{aligned}\Pr\{\mathbf{N}_e^{2\ell} \text{ is not tree like}\} &\leq \frac{c^2\frac{\overline{V}_{\ell^*}^2}{N} + \frac{\overline{C}_{\ell^*}^2}{N}}{B} \\ &= \frac{c^2\overline{V}_{\ell^*}^2 + \overline{C}_{\ell^*}^2}{NB}.\end{aligned}$$

■

If we want to run the message passing for ℓ iterations and we want $\Pr\{\mathbf{N}_e^{2\ell} \text{ is not tree like}\} \leq \epsilon$, then

$$\tfrac{c^2\overline{V}_{\ell^*}^2 + \overline{C}_{\ell^*}^2}{NB} \leq \epsilon,$$

and substituting the values $\overline{C}_{\ell^*} \approx N^{2\ell-1}$ and $\overline{V}_{\ell^*} \approx N^{2\ell}$ we get

$$\begin{gathered}\tfrac{c^2N^{4\ell}+N^{4\ell-2}}{NB} \leq \epsilon \Rightarrow \\ B \geq 2c^2\epsilon^{-1}N^{4\ell}.\end{gathered}$$

C. Importance of Jitter

The purpose of this section is to point out the importance of jitter in our model. To begin, consider a high SNR system with flat fading $h_i^c = h_i$ for all $i, c \in \{1,\ldots,N\}$ where h_i is a positive constant. That is, the fading coefficients are constant for each user and does not change over time. An indoor wireless LAN is a good example of such system where the fading coefficients are constant for each user over time.

We prove the necessity of jitter by showing that for the above system, when the maximum delay $W = 0$, the linear equations are not full rank even if the users permute their packets before each round of transmission. Observing the above while the users are not permuting their bits is trivial, since the fading coefficients are constant, and as a result the AP receives NB copies of the same equation which obviously makes matrix A singular.

Theorem 3: For the above system, if the users transmit packets consecutively without any delay ($W = 0$), the matrix A is rank deficient regardless of the particular permutations used.

Proof: Define $(bit_{n,b})$ as the matrix of bits for each user $n \in \{1, \ldots, n\}$ and each bit $b \in \{1, \ldots, B\}$. Define x_{nb} as the matrix of "*faded-bits*" where each row n is multiplied by the fading coefficient h_n (so that $x_{nb} = h_n bit_{nb}$) for all n and b). If $W = 0$, each faded-bit received on each collision round is a sum of contributions from each of the N users. Assume each user permutes its bits, and $\left(\tilde{x}_{nb}^{(c)}\right)_{b=1}^{B}$ represents the vector of permuted bits for user $n \in \{1, \ldots, N\}$ on collision round $c \in \{1, \ldots, N\}$. It is assumed that the receiver knows the permutation orders used on each transmission. Let $u_{c,b}$ represent the b^{th} bit received on the c^{th} collision round, for $b \in \{1, \ldots, B\}$ and $c \in \{1, \ldots, N\}$:

$$u_{c,b} = \sum_{n=1}^{N} \tilde{x}_{nb}^{(c)}$$

These values $u_{c,b}$ represent NB linear equations, involving NB unknowns (x_{nb}). Thus, matrix A will be rank deficient if at least one of the equations is redundant.

Let S be the sum of all elements in the $N \times B$ matrix (x_{nb}). On each collision round $c \in \{1, \ldots, N\}$, each row of matrix $(\tilde{x}_{nb}^{(c)})$ is just a permutation of the matrix (x_{nb}), and so the sum of all its elements is also equal to S, i.e.

$$\sum_{b=1}^{N} \sum_{n=1}^{N} \tilde{x}_{nb}^{(c)} = S, \quad \text{for all } c \in \{1, \ldots, N\} \quad .$$

Let us now sum the B received bits $u_{1,b}$ from the first collision round

$$S = \sum_{b=1}^{B} u_{1,b} = \sum_{b=1}^{B} \sum_{n=1}^{N} \tilde{x}_{nb}^{(1)}. \tag{2}$$

Now sum the first $B - 1$ received values $u_{2,b}$ on the second round of collision, i.e.

$$\begin{aligned} \sum_{b=1}^{B-1} u_{2,b} &= \sum_{b=1}^{B-1} \sum_{n=1}^{N} \tilde{x}_{nb}^{(2)} \\ &= \sum_{b=1}^{B} \sum_{n=1}^{N} \tilde{x}_{nb}^{(2)} - \sum_{n=1}^{N} \tilde{x}_{nB}^{(2)} \\ &= S - u_{2,B} \end{aligned}$$

Subtracting this from equation (2) yields:

$$\sum_{b=1}^{B} u_{1,b} - \sum_{b=1}^{B-1} u_{2,b} = u_{2,B} \quad .$$

Therefore, we can infer the value of the $(2B)^{th}$ received value $\mathbf{u}_{2B} = u_{2,B}$ only from observing the first $2B - 1$ bits, and hence the equation associated with this bit is redundant. ∎

Notice that, as discussed in IV-E, rank deficiency of matrix A does not mean that the received patterns are not decodable. In the high SNR regime, however, when the SNR of the senders are in the same level, ML detection simply becomes solving the linear system $A\mathbf{x} = \mathbf{u}$. If the collision pattern is decodable by ZigZag decoding, it is equivalent to Gaussian elimination (essentially only doing the back-substitution steps). Such systems are discussed in the next section.

D. Random Delay of one Symbol Suffices

We now show that, even if the users do not permute their symbols, a random delay of one-time slot $W = 1$ is sufficient for the above system to be decodable. Observe that if we can decode the first bits of all packets, then we can decode the system. The reason is that as the first bits of all packets are decodable and known, then by removing the first bits from the equations we will again have the same pattern with packets of length $B - 1$, since there are no permutations. It should be pointed out that it is easy to arrange a coordinated deterministic system with maximum delay of one symbol for it to be decodable. We assume that N users with similar SNR's want to transmit their packets to the AP and we require each of them to transmit its packet N times. At i^{th} transmission $i \in \{1, 2, \ldots, N\}$, user i transmits its packet with zero delay and rest of the users transmit with one-time slot delay. As explained, the system will be decodable.

Consider the uncoordinated model introduced in the previous section with $W = 1$. In other words, each user may transmit its packet without any delay or with the delay of one-time slot both with probability of $1/2$. In this case the decodability of the system solely depends on the uniqueness of the collision patterns. In other words, we form the $N \times N$ jitter matrix $\mathbf{J}$ which consists of the coefficients of the first equations of each collision, i.e $\mathbf{J}\left[x_{11}, \ldots, x_{N1}\right]^T = \left[u_{11}, u_{21}, \ldots, u_{N1}\right]^T$, then the pattern is decodable if the matrix $\mathbf{J}$ is non-singular. As an example, the linear equations and their resulted matrix $\mathbf{J}$ for the pattern shown in Fig. 2 are

$$\mathbf{J} = \begin{bmatrix} h_1 & 0 \\ h_1 & h_2 \end{bmatrix}, \quad \begin{cases} h_1 x_1 = u_{11} \\ h_1 x_1 + h_2 y_1 = u_{21} \end{cases}$$

Derive the matrix $\overline{\mathbf{J}}$ by factoring out h_i coefficients $i \in \{1, 2, \ldots, N\}$ from each column of matrix $\mathbf{J}$. Notice that $\overline{\mathbf{J}}$ is an $N \times N$ matrix with elements independently taking the values $(0, 1)$ with probability $1/2$, and $|\mathbf{J}| = |\overline{\mathbf{J}}| \prod_{i=1}^{N} h_i$. A classic result by Komlós [3] establishes that matrices such as $\overline{\mathbf{J}}$ are nonsingular with probability going to 1 as $N \to \infty$.

E. Importance of Fading

We note that our SigSag algorithm exploits fading in a way that pure back-substitution or Gaussian elimination cannot. To illustrate the importance of fading, we consider a simple example with 3 users and zero noise. Assume the delay offset is 0 in all three collision rounds. Thus, the resulting equations are redundant and the matrix A is obviously rank deficient. ZigZag would thus fail in this scenario. However, our algorithm would perfectly decode each user with probability 1. For intuition on this, we simply show that the bits are perfectly decodable with probability 1 just by looking at the first collision: Suppose the channel coefficients on the first round are $h = [h1, h2, h3]$. The first bit transmitted by each user is either $+1$ or -1. Thus, there are only 8 possible resulting collision values for the first received symbol:

$$\begin{aligned} &h_1 + h_2 + h_3 \\ &h_1 + h_2 - h_3 \\ &h_1 - h_2 + h_3 \\ &h_1 - h_2 - h_3 \\ -&h_1 + h_2 + h_3 \\ -&h_1 + h_2 - h_3 \\ -&h_1 - h_2 + h_3 \\ -&h_1 - h_2 - h_3 \end{aligned}$$

If all 8 possible values are distinct, then we can completely decode the first bit of all 3 users. Clearly this holds not just for the first bit, but for all bits, and so in the noiseless case with different fading coefficients, all three packets can be completely decoded after only one transmission round. It is clear that if the distribution on the joint fading coefficients have a continuous joint density, then the probability that all 8 collision values are distinct is equal to 1. For example, the probability $Pr\,[h_1 + h_2 + h_3 = h_1 + h_2 - h_3]$ is the probability that the 3-dimensional vector (h_1, h_2, h_3) lies on the 2-dimensional plane $h_3 = 0$, and hence this probability is equal to 0. While this example is for the noiseless case, it shows that fading can be exploited to achieve very high decoding probability in the high SNR regime.

V. Experimental Results

In this section we present our preliminary experimental evaluation of SigSag for $N = 2, 3$ users, and compare against a ZigZag decoder and a coordinated system that uses an interference-free round-robin TDMA schedule. Surprisingly, we find that the performance of our system greatly surpasses that of the coordinated TDMA. The reason is that each bit influences multiple observations, *acting as a code that our near-optimal decoding algorithm can decode with high probability*. Since any message passing algorithm can be used for SigSag to compute likelihoods of bits, we consider both max-product and sum-product. We call the former SigSag max-product (SSMP), and the latter SigSag sum-product (SSSP).

For our simulation we consider an additive white Gaussian noise channel with fading modeled as a uniform random variable chosen between $[0.8, 1.2]$. The block fading is assumed, i.e. fading is identical for all the bits of a packet in each transmission; but it can vary across users and in different transmissions. In our simulations, Users transmit packets of length $B = 100$, with the maximum waiting time of $W = 10$. Here, we consider the case where packets always collide with each other, and ZigZag never fails to decode, i.e. the collision patterns are not identical.

Our first experiment compares the performance of SigSag, without any permutation, versus forward ZigZag, ZigZag (forward-backward), and TDMA. For SigSag, Message passing is run for 20 iterations (after which it converged). As shown in Fig. 10, for $N = 2$ users, SSMP and SSSP performs quite similar to each other while both surpass ZigZag and TDMA with a great margin.

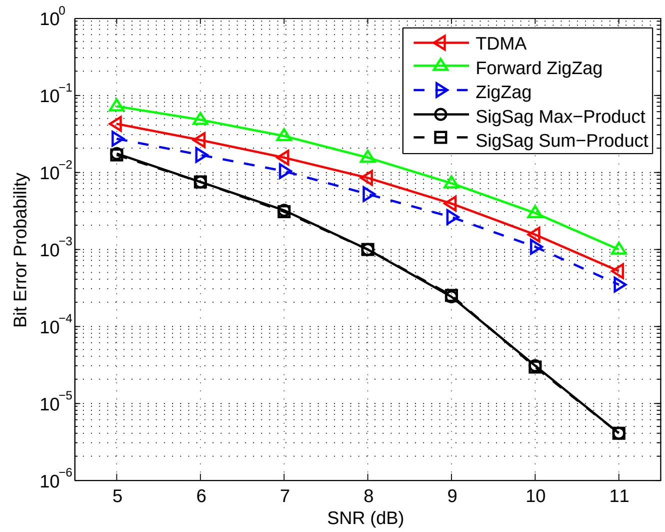

Fig. 10. Bit error rate comparison of TDMA, forward zigzag, forward-backward zigzag, Sigsag max-product, and Sigsag sum-product over different SNR's for $N = 2$ users transmitting packets of length $B = 100$.

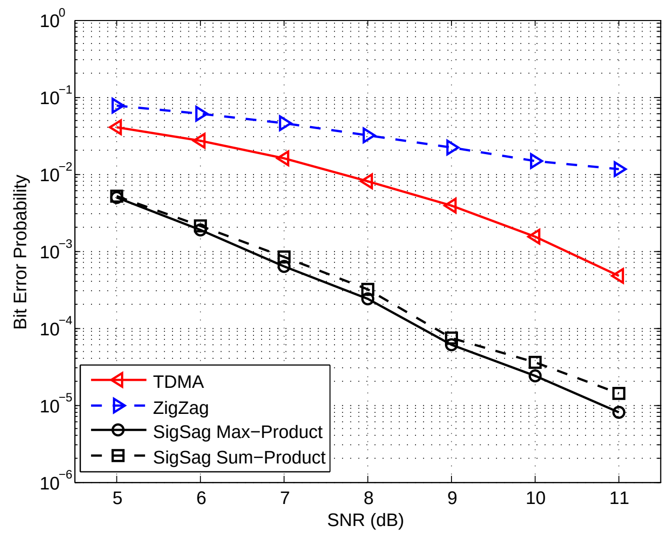

Fig. 11. Bit error rate comparison of TDMA, zigzag, Sigsag max-product, Sigsag sum-product over different SNR's for $N = 3$ users transmitting packets of length $B = 100$.

For the second experiment, we consider the previous scenario, only this time with $N = 3$ users. For simulating ZigZag, we used its extension for more than two users from [1]. As shown in Fig. 11, the performance of ZigZag degrades as number of users increase and it becomes worse than TDMA. Again, as before, SSMP and SSSP surpasses other methods while this time SSMP performs even better than SSSP.

Notice that our proposed algorithms are computationally efficient, requiring a linear number of messages per iteration and typically a number of iterations that scales logarithmically in the packet sizes and number of users [7], [22]. There is slightly more complexity compared to ZigZag since soft information needs to be communicated with each message but the performance benefits would probably justify these additional requirements for most applications.

VI. CONCLUSION

In this paper we introduced a novel decoding algorithm that is compatible with the 802.11 framework and allows the decoding of interfering users. As it is well known in the information and coding theory community, repetition codes are highly suboptimal. In this paper we show that *repetition with the addition of small random jitter and bit permutation forms good codes* that can be efficiently decoded and can outperform TDMA. There are several issues that need to be further investigated. On the practical side, how to enable bit permutations and access to soft information with current infrastructure would be important for exploiting interference at the physical layer [11], [20]. Further, our fading assumptions would need to be tested in real deployments and the performance of our algorithm investigated under different fading and retransmission models. On the theoretical side, one future direction is to use density evolution [7] to predict the exact performance of the proposed decoders and further optimize the design of the jitter distribution. More generally, investigating the factor graphs created by uncoordinated users interfering and exploiting the message-passing inference machinery for such problems is promising direction for wireless.

ACKNOWLEDGMENT

The authors would like to thank Prof. Giuseppe Caire for several stimulating discussions. This material is supported in part by one or more of the following: the DARPA IT-MANET grant W911NF07-0028, the NSF grant CCF-0964479.

REFERENCES

[1] S. Gollakota and D. Katabi, *ZigZag Decoding: Combating Hidden Terminals in Wireless Networks*, ACM SIGCOMM, vol. 38, no. 4, pp. 159–170, Oct. 2008.

[2] J. Boutros and G. Caire, *Iterative multiuser joint decoding: unified framework and asymptotic analysis*, IEEE Trans. on Info. Theory, July 2002, pp. 1772–1793.

[3] J. Komlós, *On the determinant of* $(0, 1)$ *matrices*, Studia Scientiarum Mathematicarum Hungarica, pp. 7–22, 1967.

[4] S. Verdu, *Multiuser Detection*, Cambridge, U.K.: Cambridge Univ.Press, 1998.

[5] M. Luby, *LT Codes*, The 43rd Annual IEEE Symposium on Foundations of Computer Science, 2002.

[6] F. R. Kschischang, B. J. Frey, and H. A. Loeliger, *Factor graphs and the sum-product algorithm*, IEEE Trans. Info. Theory, vol. 47, no. 2, pp. 498–519, Feb. 2001.

[7] T. Richardson and R. Urbanke, *The capacity of low-density parity check codes under message-passing decoding.* IEEE Trans. Info. Theory, vol. 47, pp. 599–618, Feb. 2001.

[8] D. Reynolds, X. Wang, and H. V. Poor, *Blind adaptive space-time multiuser detection with multiple transmitter and receiver antennas*, IEEE Trans. on Signal Processing, vol. 50, no. 6, pp. 1261–1276, June 2002.

[9] H. El Gamal and E. Geraniotis, *Iterative multiuser detection for coded CDMA signals in AWGN and fading channels*, IEEE Journal on Selected Areas in Communications, vol. 18, pp. 30–41, Jan. 2000.

[10] H. V. Poor, *Iterative multiuser detection*, IEEE Signal Processing Magazine, Vol. 21, pp. 81–88, Jan. 2004.

[11] L. Erran, J. Liu, Kun Tan, H. Viswanathan, and R. Yang, *Retransmission Repeat: Simple Retransmission Permutation Can Resolve Overlapping Channel Collisions*, Eighth ACM Workshop on HotNets, Oct. 2009.

[12] B. Nazer and M. Gastpar, *Compute-and-forward: Harnessing interference with structured codes*, *submitted to IEEE Trans. Info. Theory*. Available under: http://arxiv.org/abs/0908.2119

[13] X. Zhang, K. G. Shin, *Chorus: Collision Resolution for Efficient Wireless Broadcast*, Proceedings of IEEE INFOCOM, 2010.

[14] J. G. Andrews, *Interference Cancellation for Cellular Systems: A Contemporary Overview*, IEEE Wireless Communications Magazine, Apr. 2005.

[15] A. Kamra, J. Feldman, V. Misra, and D. Rubenstein, *Growth Codes: Maximizing Sensor Network Data Persistence*, Proceedings of ACM SIGCOMM, Pisa, Italy, Sep. 2006.

[16] F. Soldo, A. Markopoulou, and A. Lopez-Toledo, *A Simple Optimization Model for Wireless Opportunistic Routing with Intra-session Network Coding*, in Proc. of NetCod 2010, Toronto, Canada, June 2010.

[17] A. Khreishah, C. C. Wang, and N. B. Shroff, Cross-layer Optimization *for Wireless Multihop Networks with Pairwise Intersession Network Coding*, IEEE Journal on Selected Areas in Communications, vol. 27, no. 5, pp. 606–621, June 2009.

[18] A. Eryilmaz, A. Ozdaglar, and M. Medard, *On the Delay and Throughput Gains of Coding in Unreliable Networks*, IEEE Trans. on Info. Theory, vol. 54, no. 12, pp. 5511–5524, 2008.

[19] H. Pishro-Nik, N. Rahnavard, and F. Fekri, *Non-uniform error correction using low-density parity check codes*, IEEE Trans. on Info. Theory, vol. 51, no. 7, pp. 2702–2714, July 2005.

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

[21] S. Y. El Rouayheb, A. Sprintson, and C. Georghiades, *On the Relation Between the Index Coding and the Network Coding Problems*, in proceedings of ISIT 2008, Toronto, Canada.

[22] T. Richardson and R. Urbanke, *Modern Coding Theory*, Cambridge University Press, 2008.

[23] J. Paek and M. J. Neely, *Mathematical Analysis of Throughput Bounds in Random Access with ZigZag Decoding*, Proc. of 7th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt), June 2009.