

Implicit Transmission of Coded Information

RANA A. HASSAN^{id} AND JOHN P. FONSEKA^{id}

Department of Electrical and Computer Engineering, The University of Texas at Dallas, Richardson, TX 75080, USA

CORRESPONDING AUTHOR: R. A. HASSAN (e-mail: rana.hassan@utdallas.edu)

ABSTRACT A novel implicit transmission with bit flipping (ITBF) technique is introduced to transmit a coded stream implicitly while transmitting a coded stream explicitly over a channel. ITBF flips a set of chosen parity bits of the explicitly transmitted stream according to an implicit stream. Numerical results obtained using the low density parity check (LDPC) code employed in the 5G standard show that ITBF can transmit an implicit stream up to 13.19% of the rate of transmission of the explicit stream without significantly sacrificing performance, or increasing the decoding complexity or the decoding delay. The ITBF is combined with collection of punctured code decoding (CPCD) to form implicit transmission with collection decoding (ITCD) schemes that can further increase the rate of transmission on the implicit stream without increasing the decoding delay, however, with a slight increase in the decoding complexity. It is demonstrated with the LDPC code in the WiFi standard that ITCD can transmit an implicit stream at up to 25% of the rate of transmission of the explicit stream.

INDEX TERMS LDPC codes, implicit transmission, rate enhancement.

I. INTRODUCTION

ALMOST all current communication systems employ some form of error control coding to improve the reliability of transmission. LDPC codes, turbo codes, polar codes, etc. are commonly used in current systems [1], [2], [3], [4]. The studies so far have focused primarily on searching for good coding techniques and searching for good high-rate codes within those coding techniques [2], [5]. Studies have also focused on improving the decoding of coded signals to achieve good performance with low decoding complexity and decoding delay [6], [7].

Instead of the traditional methods of searching for good codes, it is highly desirable to be able to transmit coded bits implicitly, without transmitting them physically over a channel, while transmitting a coded stream explicitly over that channel. It would be highly beneficial if such schemes can be developed without increasing the decoding complexity or the decoding delay while maintaining a significant data rate on the implicitly transmitted stream. If such implicit transmission methods can be developed, they can preferably be used with known powerful codes that have been derived using traditional methods.

In this study, we present such a scheme, referred to as implicit transmission with bit flipping (ITBF), to transmit a second coded stream (referred to here as the secondary

stream or the implicit stream) implicitly without physically transmitting it over a channel during the transmission of a first coded stream (referred to here as the primary stream or the explicit stream) over that channel. In this study, we present a simple way to transmit a secondary stream implicitly without significantly sacrificing the performance of the primary stream or increasing the decoding delay.

In [8], [9], implicit transmission has been proposed to transmit a turbo-coded implicit stream while transmitting a turbo-coded stream explicitly. To the best of our knowledge, [8] is the first publication that presents a technique to transmit a coded stream implicitly while transmitting a coded stream explicitly. The other known techniques that use implicit transmission, such as spatial modulation (SM) [10] and index modulation (IM) [11], [12]. SM adds a new spatial domain using additional antennas to transmit additional information implicitly, while IM adds indices to the transmitted symbols to transmit additional information bits. Even though schemes presented in [8], [9] can transmit a stream implicitly, the decoding of the two streams need to be done jointly by running iterations between the explicit stream and the implicit stream. As a result, the receiver proposed in [8] increases the decoding complexity and the decoding delay significantly. Specifically, the schemes presented in [8], [9] demand an increase in the decoding complexity

and the decoding delay by a factor of at least 6 to 10 due to the transmission of the implicit stream. Further, the schemes presented in [8], [9] are: (a) dependent on the coding technique; it generates attractive schemes with turbo-coded systems but fails with codes such as LDPC codes, (b) requires use of codes that are similar in coding power on the explicit and implicit streams due to the exchange of information between them during decoding.

In this study, we describe a general ITBF method to transmit a secondary coded stream implicitly during the transmission of a primary coded stream explicitly. Throughout this paper, the primary stream is also referred to as the explicit stream while the secondary stream is also referred to as the implicit stream. In contrast to the schemes in [8], [9], the proposed ITBF method here treats the explicit and the implicit streams independently. Therefore, ITBF does not require iterations that involve both the explicit and implicit streams as in [8], [9], and as a result, does not increase the decoding delay or significantly increase the decoding complexity. In particular, our contributions in this work are as follows.

- We introduce ITBF method that transmits an explicit coded stream over a channel by performing close to the performance that can be achieved by transmitting that explicit coded sequence without any flips.
- We show that ITBF simultaneously allows the transmission of an independent implicit coded bit stream without transmitting any of its coded bits over the channel while maintaining better performance on that implicit stream than that on the explicit stream, and maintaining a noticeable fraction of the transmission rate of the explicit stream on that implicit stream.
- We demonstrate that ITBF schemes are further improved by combining them with collection of punctured code decoding (CPCD), presented in [13], [14] and summarized later in Section V, to form a new class of hybrid ITBF/CPCD schemes which are simply referred to as implicit transmission with collection decoding (ITCD) schemes.
- We demonstrate that ITCD schemes maintain better performance both on the explicit and implicit streams, and maintain a higher data rate on the implicit stream compared to ITBF schemes.
- Simulation results presented later in Section V justify all contributions listed above related to both ITBF and ITCD.

The rest of the paper is organized as follows. Section II describes the proposed ITBF signaling method. Decoding of explicit and implicit streams are discussed in Section III. Section IV discusses the proposed hybrid ITBF/CPCD (ITCD) method. Section V presents simulation results and discusses the potential applications of the proposed schemes. The paper is concluded in Section VI.

II. ITBF METHOD

In order to describe the ITBF technique, let us consider a code that generates n coded bits corresponding to every k

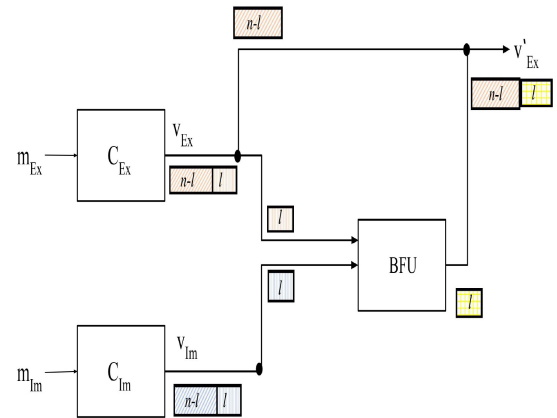


FIGURE 1. General structure of an ITBF transmitter.

message bits, where $k < n$. Then it is possible to choose $l (< (n - k))$ bits out of $(n - k)$ parity bits that can be removed from the coded sequence and yet correctly recover the original message sequence of length k . These l bits can preferably be chosen by using a good known punctured code generated from code C . For example, let us consider a code C with rate $\frac{1}{3}$, i.e., $n = 3k$. Then consider a rate $\frac{1}{2}$ punctured code generated from that rate $\frac{1}{3}$ code C . Note that in the construction of the rate $\frac{1}{2}$ punctured code, $n/6$ coded bits of C are identified and removed. Hence, these $n/6$ coded bits can be selected as the $l = n/6$ coded bits of C . Of course, depending on the selected punctured code, the set of l coded bits and its length can change. Throughout this study, these l bits are referred to as the chosen bits. Therefore, a set of chosen bits of a code C can be pre-selected by preferably considering a punctured code generated from the code C or by using any other method.

Fig. 1 illustrates the structure of the transmitter for the proposed ITBF scheme. In ITBF, an explicit message sequence m_{Ex} and an implicit message sequence m_{Im} are separately encoded according to two codes C_{Ex} and C_{Im} to generate two coded streams v_{Ex} and v_{Im} respectively. Without loss of generality, in this discussion, we consider both C_{Ex} and C_{Im} to be the same code C , i.e., $C_{Ex} = C_{Im} = C$. However, as stated before, C_{Ex} and C_{Im} can be two independent codes. Then identify the l number of pre-selected chosen bit positions of v_{Ex} . Then, according to l bits of the coded implicit stream v_{Im} , flip the chosen bits of explicit coded stream v_{Ex} using a bit flipping unit (BFU) as illustrated in Fig. 1. Specifically, BFU flips each of these chosen bits of v_{Ex} if the corresponding coded bit of the implicit stream is a 1 (or a 0) and not flipped if the corresponding coded implicit bit is a 0 (or a 1). Fig. 2 illustrates an example of the flipping operation performed by the BFU when $l = 6$. The resulting sequence v'_{Ex} on the explicit stream is then transmitted over the channel. Note that (a) the transmitted sequence v'_{Ex} and the coded sequence v_{Ex} have the same length n , (b) none of the coded bits of v_{Im} are transmitted over the channel, however, the effect of the corresponding l

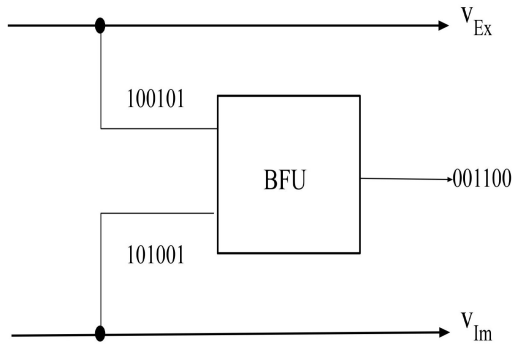


FIGURE 2. Illustration of the actions taken by BFU when $l = 6$.

bits of v_{Im} that determined the flipping of the l chosen bits of v_{Ex} , is transferred by v'_{Ex} , and (c) following (a) and (b), the information of the explicit stream contained in v_{Ex} and the information of the corresponding l coded bits of v_{Im} are carried by v'_{Ex} . Therefore, upon transmitting $\lceil n/l \rceil$ number of blocks of v'_{Ex} , information of all coded bits of one full codeword of v_{Im} will be available allowing the receiver to decode a codeword of v_{Im} . Note also that, due to the flipping of the chosen bits, the transmitted sequence v'_{Ex} may very well not be a valid coded sequence of C_{Ex} . It is mentioned here that bit flipping is known in the literature as a decoding method for LDPC codes similar to the min-sum decoding algorithm; see bit-flipping decoding [15] and min-sum decoding [16], [17]. However, bit flipping considered here is completely different as it is used at the transmitter to incorporate an implicit stream, whereas bit flipping discussed in the literature [15] deals solely with decoding.

III. DECODING OF EXPLICIT AND IMPLICIT STREAMS

The proposed ITBF receiver constructed here is based on the following observation: Even though the transmitted sequence v'_{Ex} may not be a valid codeword of C_{Ex} , any existing invalidity is caused only within the l chosen bits due to the flipping of the bits that occurred prior to transmission. Therefore, if the received signal is initially decoded as a punctured code by ignoring those l chosen bits, the explicit message sequence m_{Ex} and the corresponding explicit coded sequence v_{Ex} can be correctly decoded. Further, this decoding provides information about the l chosen bits of v_{Ex} (without any flips) while the received signal provides information of v'_{Ex} (with the flips).

Fig. 3 shows the general structure of the ITBF decoder proposed here based on the above observation to recover both the explicit and the implicit streams. Below we explain the steps involved in ITBF decoding. Throughout this discussion, we focus primarily on the l chosen bits. In order to assist that discussion, we denote the following quantities of the l chosen bits:

- (a) received signal values by y_1, y_2, \dots, y_l ,
- (b) coded sequence values of v_{Ex} (prior to flipping) by $v_{Ex_1}, v_{Ex_2}, \dots, v_{Ex_l}$, and
- (c) transmitted sequence v'_{Ex} (after flipping) by $v'_{Ex_1}, v'_{Ex_2}, \dots, v'_{Ex_l}$.

A. DECODING PROCEDURE

The decoding procedure is described using the following four steps.

- 1) *Initial Decoding*: Initially, decode C as a punctured code by removing the l chosen bits from the received signal. If iterative decoding is used (such as in the decoding of a LDPC code), run initially a set of iterations of the punctured code. The number of initial iterations used for the punctured code can be pre-selected or adaptively varied depending on the signal to noise ratio or as the iterations progress. Note that the initial decoding also provide likelihood values of the l chosen bits, which are denoted here by $L_{v_{Ex(i)}}$, $i = 1, 2, \dots, l$, and they represent the likelihood values of the encoded sequence v_{Ex} prior to flipping at the transmitter.
- 2) *Detecting Flips*: In order to decide whether or not each of the l chosen bits is flipped, hard decode each of the l chosen bits of v_{Ex} , which are denoted by b^i , $i = 1, 2, \dots, l$, using the likelihood values of those bits found in step 1. Additionally, for each of the l chosen bits, hard decode the corresponding received signal value y_i , $i = 1, 2, \dots, l$, to determine the hard decoded received signal value, y'_h , $i = 1, 2, \dots, l$ that correspond to the flipped sequence v'_{Ex} . Then, using b^i and y'_h values, determine f_i as

$$f_i = \begin{cases} 0 & b^i = y'_h \\ 1 & b^i \neq y'_h \end{cases} \quad (1)$$

to indicate whether the i th bit was likely to have been flipped or not prior to transmission. Specifically, if $f_i = 0$, the i th chosen bit is not likely to have been flipped while if $f_i = 1$, it is likely that the i th bit has been flipped. Note that f_i values, which use the hard decoded received signal, are corrupted by channel noise similar to a received signal.

- 3a) Since f_i , $i = 1, 2, \dots, l$, found in step 2, indicates whether or not the i th chosen bit has been flipped, it can then be used to modify the received sequence to correspond to v_{Ex} by reversing the effect of the flips for the decoding of the explicit stream. Specifically, the received signal y_i can be corrected for the l chosen bits in the decoding of the explicit stream as $(1 - 2f_i)y_i$, $i = 1, 2, \dots, l$.
- 3b) In case of iterative decoding, upon reversing the effects of the flips, continue decoding of C as a full code (not as a punctured code) by also including the corrected received signal values of the chosen bits.
- 4) Recall that the flipping of the l chosen bits were decided at the transmitter according to the implicit coded stream. Further, observe that if the i th implicit bit, Im_i , had actually been transmitted over the channel with the same noise experienced by y_i , it would have been received as $(1 - 2Im_i)|y_i|$.¹ Hence, an artificially created received

1. It is assumed here that a positive signal value is used for $Im_i = 0$ and a negative signal value is used for $Im_i = 1$.

signal value can be obtained for each implicit coded bit Im_i , $i = 1, 2, \dots, l$ as $(1 - 2f_i)|y_i|$. Note that depending on the value of f_i (0 or 1), the artificially created channel value is positive or negative suggesting that the i th coded implicit bit has not been flipped or flipped respectively. Even though f_i values are available after the initial decoding, a more reliable set of f_i values in (1) can be calculated by using the $L_{V_{Ex(i)}}$ values at the end of the decoding of the explicit stream in step 3b. These re-calculated f_i values can be used to calculate the artificially created received signal values for the corresponding l coded implicit bits. Since f_i decisions are noisy, the artificially created received signal values of the Im_i , $i = 1, 2, \dots, l$, are also noisy similar to channel information extracted from a noisy received signal. Therefore, in order to maintain good performance on the implicit stream, it is necessary to employ an error control code on the implicit stream similar to the explicit stream.

The first three steps describe the decoding of a single codeword of the explicit stream. It is noticed that if the initial decoding step 1 and the calculation of f_i s in step 2 are reliable then step 3a would provide a reliable explicit stream that would perform almost as reliably as if no bits were flipped prior to transmission. It is also seen that when every codeword of the explicit stream (n coded bits) is transmitted over the channel, an artificially created channel information of l coded bits of the implicit stream can be extracted without transmitting any of those l bits over the channel.

B. DECODING STRATEGY

If C is a small code, it can be decoded in a maximum likelihood (ML) sense in the initial decoding of C in step 1 and the decoding of C as a full code in step 3. However, this doubles the decoding complexity and the decoding delay of the explicit stream.

For a large code, such as an LDPC code and most other codes used in practice, ML decoding is not possible and instead iterative decoding is commonly used. In such situations, the initial decoding in step 1 and the full decoding in step 3b can be done in an efficient manner without increasing the overall decoding delay or the decoding complexity. Throughout this study, the decoding delay is measured in terms of the total number of iterations while the decoding complexity is measured by the number of times the SPA algorithm is called during decoding thereby disregarding any delay or complexity added by the calculation of f_i s and the correction of the flipped bits of the explicit stream. Focusing on the iterative decoding of C , a pre-selected N_1 number of iterations in step 1 and a pre-selected N_2 number of iterations in step 3b can be used. The values of N_1 and N_2 can be chosen to maintain the total number of iterations $N = (N_1 + N_2)$ close to the number of iterations commonly used without ITBF thereby maintaining about the same decoding delay and the decoding complexity.

IV. HYBRID ITBF/CPCD (ITCD) SCHEMES

Collection of punctured code decoding (CPCD) has been recently introduced to improve the decoding of a code [13], [14]. It has been shown that CPCD can significantly improve the performance of quasi-cyclic LDPC (QC-LDPC) codes [13], [14]. In this section, we explain how ITBF can be combined with CPCD to generate ITCD schemes that can transmit a higher data rate on the implicit stream than using ITBF alone while also improving performance on both explicit and implicit streams.

A. REVIEW OF CPCD [13]

In CPCD, a code C (which is considered as the mother code) is viewed as a collection of a pre-selected number of D punctured codes, C_i , $i = 1, 2, \dots, D$, generated from that mother code C [13]. Considering C in systematic form, all n coded bits are viewed as a collection of the message bits and the set of its parity bits p . In CPCD, each punctured code C_i is constructed from all message bits and a portion of the parity bits p_i , $i = 1, 2, \dots, D$. In CPCD, p_i s are formed by dividing all parity bits p into non-overlapping segments so that $\cup p_i = p$. During decoding, each C_i is separately decoded by using the received signal corresponding to its own coded bits (message portion and its corresponding parity portion p_i) and also using the extrinsic information of all bits of C_i provided by the remaining punctured codes, C_j , $j = 1, 2, \dots, D$, $i \neq j$.

B. ITCD SCHEMES

It is seen from Sections II and III that ITBF uses a punctured code in the initial decoding (step 1). Therefore, the initial decoding in ITBF inherently consists of the following two punctured codes of C : (a) the punctured code used in the initial decoding (say C_1), and (b) the code formed by the message bits and the chosen bits that are not used in the initial decoding (say C_2). However, C_2 becomes available after performing decoding steps 1, 2 and 3a. Upon determining C_2 , the decoding in ITBF was continued by considering C as a full code. Instead, decoding can be continued as in CPCD by considering C_1 and C_2 as two punctured codes. An ITBF scheme that switches to CPCD decoding after step 3a is considered as a hybrid ITBF/CPCD scheme or simply as an ITCD scheme. The block diagram of an ITCD decoder will be very similar to the ITBF decoder shown in Fig. 3 with the following two changes: (a) the “Decode C_{Ex} as a punctured code” block changed to “Decode C_1 ”, and (b) “Decode C_{Ex} as a full code” block changed to “Employ parallel CPCD with C_1 and C_2 as punctured codes”.

However, the CPCD technique considered in ITCD has differences between the CPCD technique discussed in [13], [14]. In order to elaborate on the differences, let us first recall that all CPCD schemes considered in the numerical results in [13], [14] have used the same number of parity bits in all of their punctured codes and they all started to decode from the very first CPCD iteration. However, in ITBF, C_2 becomes available for decoding only after N_1 iterations to

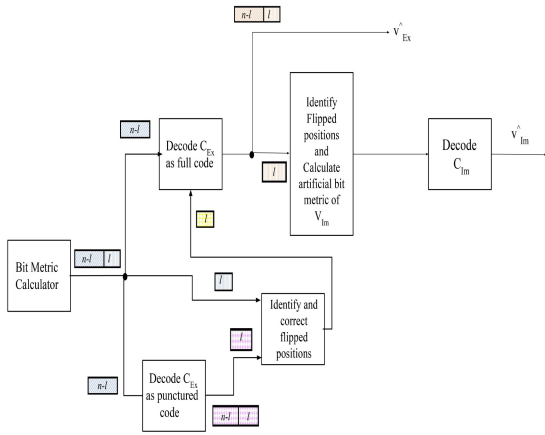


FIGURE 3. General structure of the ITBF decoder.

complete steps 1 through 3a, and further, the number of parity bits of C_2 is generally smaller than that of C_1 . Therefore, in order to employ CPCD in ITBF, we first introduce three separate special types of CPCD as follows:

- (a) Unbalanced CPCD (U-CPCD) that employs different numbers of parity bits in different punctured codes.
- (b) Staggered CPCD (S-CPCD) that starts decoding different punctured codes at different numbers of CPCD iterations.
- (c) Unbalanced-staggered CPCD (US-CPCD) is a hybrid of U-CPCD and S-CPCD that employs different numbers of parity bits and starts decoding different punctured codes at different numbers of CPCD iterations.

In ITBF, the CPCD employed is the US-CPCD type of CPCD. Therefore, ITBF schemes switch to US-CPCD after completing steps 1 through 3a in the decoding of the first punctured code C_1 . All ITCD schemes presented in this study employ parallel CPCD. Further, in contrast to the parallel CPCD presented in [13] that can reduce the decoding delay and maintain the same decoding complexity, parallel CPCD that maintains the same delay as normal SPA decoding is used here to achieve improved performance. Therefore, ITCD schemes presented here maintain about the same decoding delay but slightly increase the decoding complexity due to the decoding of both C_1 and C_2 simultaneously during the CPCD portion of decoding. If the number of iterations in the initial decoding is 8 with a total of 24 iterations, the increase in the decoding complexity will be about 67%, however, without an increase in the decoding delay. In general, if an ITCD scheme employs CPCD with D number of parallel punctured codes with N_1 number of initial decoding iterations followed by N_2 number of CPCD iterations, the decoding complexity would be increased, compared to SPA decoding, by a factor of $(N_1 + DN_2)/(N_1 + N_2)$, however, without any increase in decoding delay.

V. SIMULATION RESULTS AND DISCUSSION

A. BIT ERROR RATE (BER) SIMULATIONS

In this section, we present numerical results to demonstrate the following:

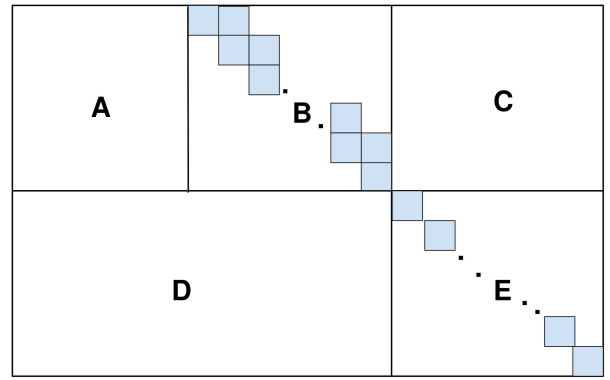


FIGURE 4. The parity check matrix H for 5G NR LDPC codes.

- (a) ITBF can transmit a separate implicit coded stream without sacrificing any significant performance on both the explicit and implicit streams.
- (b) ITCD can transmit a higher data rate on the implicit stream while maintaining the same or better performance on both the explicit and implicit streams compared to traditional decoding of the explicit stream without any implicit stream.

Further, the numerical results demonstrate that additional implicit stream can be transmitted without any increase in the decoding complexity and decoding delay in ITBF schemes. It is also demonstrated that the ITCD schemes, compared with ITBF schemes, can transmit an implicit stream at a higher data rate while also performing better on both explicit and implicit streams without increasing decoding delay, however, with a slight increase in the decoding complexity.

In order to demonstrate (a), we consider the LDPC code employed in the 5G NR standard. In the LDPC code employed in 5G NR, it is easier to identify the chosen bits needed in ITBF due to its structure. Specifically, the parity check matrix H of the LDPC code in 5G NR has a sub-block structure, shown in Fig. 4, where, each sub-block is a $z \times z$ matrix that is either an all-zero matrix or an identity matrix, and the value of z is chosen depending on the application. The segment A corresponds to the message bits. Segment B corresponds to the first set of parity bits; its first column has weight 3, while its other columns have a dual diagonal structure. Both segments A and B together represent the highest code rate that can be realized in 5G NR LDPC code. Segment C is an all-zero matrix. Segment D is called an extension region and its main purpose is to support Incremental Redundancy Hybrid Automatic Repeat Request (IR-HARQ). Segment E is an identity matrix. In 5G NR, two separate versions of the LDPC code with the same structure shown in Fig. 4 are employed; base graph 1 that employs a 46 by 68 matrix and base graph 2 that employs a 42 by 52 matrix in terms of sub-blocks [18]. In 5G NR, the rate adjustment is done by shortening the H matrix by simply removing sub-blocks along columns from the identity portion E (starting from the right most column). Therefore, puncturing in 5G NR can be easily done by puncturing the

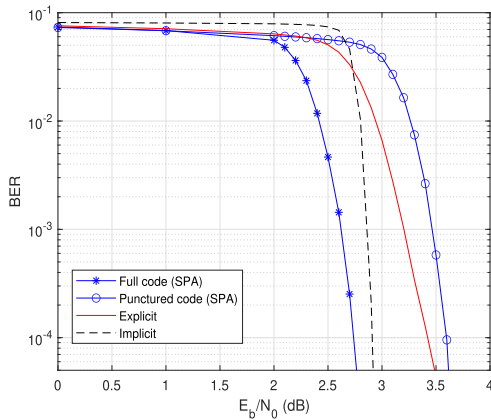


FIGURE 5. BER variations of the explicit and the implicit streams of the ITBF schemes constructed from the rate 0.2083 code that transmits 13.19% rate on the implicit stream compared to the transmission rate of the explicit stream.

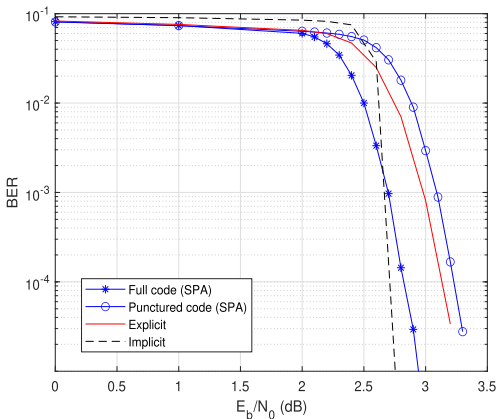


FIGURE 6. BER variations of the explicit and the implicit streams of the ITBF schemes constructed from the rate $\frac{1}{4}$ code that transmit 5.3% rate on the implicit stream compared to the transmission rate of the explicit stream.

desired number of sub-blocks (which determine the chosen bits in ITBF) from the end of the codeword.

Figs. 5 and 6 show the BER variations of the explicit and the implicit streams of two separately constructed ITBF schemes starting from the 5G LDPC code that uses the base graph 2 and employs 16-QAM for transmission. The ITBF scheme in Fig. 5 uses $z = 256$ and punctures out two sub blocks from the right of \mathbf{H} to result in a 5G code C with rate 0.2083. Further, the ITBF scheme considered in Fig. 5 that employs the above \mathbf{H} matrix uses $\frac{1}{6}$ fraction of its parity bits of the code as chosen bits thereby maintaining a 13.19% transmission rate on the implicit stream compared with the transmission rate of the explicit stream. Similarly, the ITBF scheme in Fig. 6 also uses $z = 256$ and punctures out twelve sub blocks from \mathbf{H} to result in a code C with rate $\frac{1}{4}$. The ITBF scheme considered in Fig. 6 employs that \mathbf{H} matrix and uses $\frac{1}{14}$ fraction of parity bits of the code as chosen bits thereby maintaining a 5.3% transmission rate on the implicit stream compared with the transmission rate of the explicit stream. The ITBF schemes in Figs. 5 and 6 use 8 initial iterations and 24 total iterations in their BER variations. The

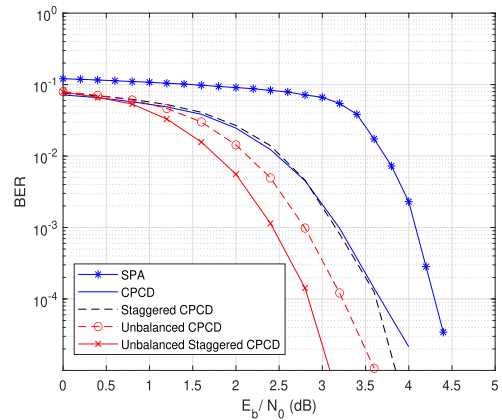


FIGURE 7. Comparison of B-CPCD, U-CPCD, S-CPCD and US-CPCD schemes constructed from the rate $\frac{1}{2}$ code with 16-QAM transmission.

BER variations of the ITBF schemes are compared with the full code and the punctured code (used in the initial decoding) with the same number of 24 iterations. It is seen from Figs. 5 and 6 that the explicit stream of the ITBF schemes can perform better than the BER variation of the punctured in isolation (demonstrating that the proposed identification of the flipped positions and the correction of the received signal values are indeed helping the decoder), and further, the BER variation of the explicit stream gets closer to the SPA decoding of the full-code. It is also seen that the BER variation of the implicit stream is better than that of the explicit stream and close to that of the BER variation of the full code. Thus, the proposed ITBF technique can transmit a secondary coded stream implicitly without noticeably sacrificing performance while maintaining the same decoding delay and the decoding complexity which are measured in terms of the total number of iterations.

Since ITCD employs CPCD, and it is known that QC-LDPC codes perform well with CPCD [13], [14], we consider the QC-LDPC code with the length 1944 employed in the WiFi standard [19] in all numerical results related to ITCD. Since ITCD employs US-CPCD, we first compare the BER variation of the known CPCD technique in [13], that employs balanced distribution of parity bits among punctured codes and called here as the balanced CPCD (B-CPCD), with those of U-CPCD, S-CPCD and US-CPCD with two punctured codes ($D = 2$). All U-CPCD schemes considered here employ 75% of parity bits in the first punctured code C_1 and the remaining 25% of parity bits in the second punctured code C_2 . In all S-CPCD schemes the first punctured code C_1 runs 8 iterations before CPCD decoding begins with a total of 24 iterations. Fig. 7 shows the BER variation of B-CPCD, U-CPCD, S-CPCD, and US-CPCD in the decoding of rate $\frac{1}{2}$ (972, 1944) LDPC code in WiFi when the symbols are transmitted using 16-QAM. Fig. 8 shows similar results when the code rate is $\frac{2}{3}$ and 16-QAM is used for transmission. It is seen from Figs. 7 and 8 that US-CPCD schemes can actually perform better than all other CPCD counterparts.

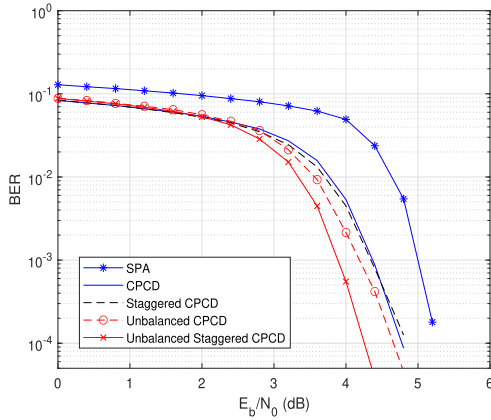


FIGURE 8. Comparison of B-CPCD, U-CPCD, S-CPCD and US-CPCD schemes constructed from the rate $\frac{2}{3}$ code with 16-QAM transmission.

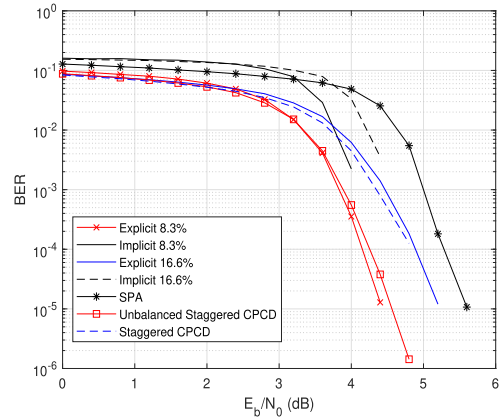


FIGURE 10. BER variations of the explicit and the implicit streams of the ITCD schemes constructed from the rate $\frac{2}{3}$ code that transmits 8.3% and 16.6% rate on the implicit stream compared with the transmission rate of the explicit stream.

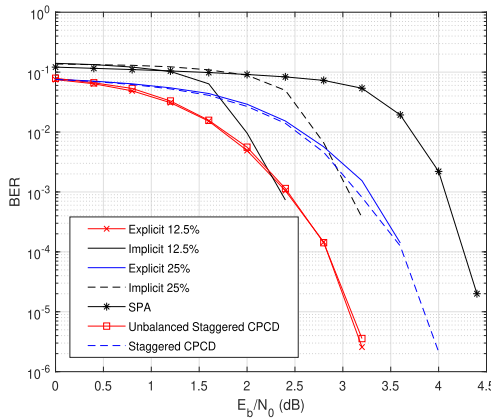


FIGURE 9. BER variations of the explicit and the implicit streams of the ITCD schemes constructed from the rate $\frac{1}{2}$ code that transmits 12.5% and 25% rate on the implicit stream compared with the transmission rate of the explicit stream.

All ITCD schemes considered here employ 8 initial iterations followed by US-CPCD with 2 punctured codes ($D = 2$) with 4 CPCD iterations and run 4 sub-iterations in each punctured code in every CPCD iteration. Therefore, all ITCD schemes considered here have the same decoding delay compared with regular SPA decoding with 24 iterations but increases the decoding complexity by about 67% due to the use of 2 punctured codes during the US-CPCD portion of the decoding. Fig. 9 shows the BER variations of the explicit and the implicit streams separately of two ITCD schemes generated from the rate $\frac{1}{2}$ LDPC code employed in WiFi when 16-QAM is used for transmission. Specifically, the two ITCD schemes maintain 12.5% and 25% transmission rate on the implicit stream compared with the transmission rate of the explicit stream. The ITCD scheme that transmits 12.5% rate on the implicit stream employs US-CPCD with 75%, 25% split of parity bits between the two punctured codes C_1 and C_2 . Similarly, the ITCD scheme that transmits 25% rate on the implicit stream employs US-CPCD with 50%, 50% split of parity bits between C_1 and C_2 (essentially making it similar to S-CPCD). Fig. 10 shows similar BER variations of two ITCD schemes generated from the

rate $\frac{2}{3}$ code employed in the WiFi standard with 16-QAM transmission. Since the number of parity bits are smaller in the rate $\frac{2}{3}$ code, ITCD schemes presented in Fig. 10 respectively transmit a lower rate, specifically 8.3% and 16.6%, on the implicit stream compared with the rate on the explicit streams. It is seen that Fig. 5 and Fig. 6 verify the claim made in Section I about the contribution that ITBF can make. Similarly, Fig. 9 and Fig. 10 verify the claim that was made in Section I about the contribution that ITCD can make.

B. APPLICATIONS OF ITBF AND ITCD

ITBF and ITCD can be employed in any communication system to improve the overall transmission rate by transmitting a secondary coded stream implicitly. It is seen from the BER variations in Section V-A that ITBF schemes can transmit a secondary coded stream implicitly without noticeably sacrificing performance while maintaining the same decoding delay and the decoding complexity. ITCD achieves the same goal as ITBF, but with a slight increase in decoding complexity. However, ITCD can maintain a higher data rate on the implicit stream while maintaining better or similar performance on both the implicit and explicit streams.

Since both explicit and implicit streams of both ITBF and ITCD schemes operate independently, different types of codes, different code rates, and desired BER values can be independently employed on the two streams. However, based on the common approach of identifying the chosen bits using a punctured code of the mother code, the proposed ITBF and ITCD techniques are more suitable for codes that can generate powerful punctured codes. However, it is important to note that codes employed in most applications, such as 5G NR and WiFi, have a rate adjustment feature which is usually implemented by puncturing the lowest rate code that has been chosen for the application. Therefore, codes employed in most applications are known to have powerful punctured codes. ITBF and ITCD techniques are highly attractive for multimedia applications. In multimedia applications, the

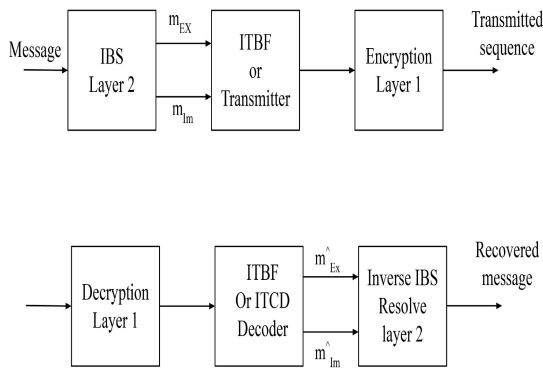


FIGURE 11. Encoding and decoding of two layer encrypted message.

explicit stream and the implicit stream can represent two different types of multimedia streams. For example, the explicit stream could transmit a video signal while the implicit stream transmits an audio signal thereby eliminating the need for a separate channel for the transmission of the audio signal.

Another important application of ITBF and ITCD is in information security. Different types of encryption methods are used in information security [20]. The transmission of an independent implicit stream in ITBF and ITCD allows a secure communication system to add an additional layer of encryption through ITBF or ITCD. Fig. 11 illustrates how an additional layer of security can be embedded while transmitting an encrypted message. This can be achieved by forming the implicit stream by choosing bits from the original message stream according to a confidential implicit bit selection (IBS) process as illustrated in Fig. 11. The traditional form of encryption can be employed on the explicit stream which is referred to here as the first layer of encryption. The second layer of encryption is added from the confidential IBS process. Note that, even if an intruder somehow penetrates the first layer of encryption and recovers the explicit information stream, the intruder is still unable to recover the original message stream without the knowledge of the IBS process. Therefore, the second layer of encryption introduced by the IBS process in the selection of the implicit message stream can significantly enhance the security of an information transmission system.

VI. CONCLUSION

A novel implicit transmission with bit flipping (ITBF) technique has been introduced to transmit a secondary coded stream implicitly while transmitting a primary coded stream explicitly over a channel. The ITBF technique can employ any coding technique on the primary and secondary channels independently. ITBF flips a set of chosen parity bits of the explicit stream according to the implicit stream. Using an initial decoding method that excludes the chosen bits, the receiver (a) corrects whether or not each of the chosen bit

has been flipped, (b) corrects the portion of the chosen bits of the explicit received stream (as if no bits were flipped before transmission) to continue decoding of the explicit stream, and (c) extracts artificial channel information for the corresponding implicit bits. Numerical results presented for the LDPC code employed in 5G NR demonstrate that the implicit stream, that employs the same code employed on the explicit stream, can transmit up to 13.19% of the rate of the explicit stream on the implicit stream without significantly sacrificing performance, or increasing the decoding complexity or the decoding delay.

The ITBF method has been further improved by combining it with CPCD [13] to form ITCD schemes. ITCD schemes move to CPCD upon correcting the flipped bits. Specifically, CPCD decoding in ITCD considers the following two punctured codes: (a) the message and the parity bits excluding the chosen bits, and (b) the message and the corrected chosen bits. Since CPCD performs better than traditional decoding, ITCD can transmit a higher data rate on the implicit stream while maintaining better or the same performance as traditional decoding without increasing the decoding delay, however, with a slight increase in the decoding complexity. Numerical results presented for the LDPC code in the WiFi standard show that ITCD can transmit a secondary stream implicitly with up to a 25% rate of the explicit stream on the implicit stream without sacrificing performance, or increasing the decoding delay but with a slight increase in the decoding complexity.

REFERENCES

- [1] W. E. Ryan et al., *An Introduction to LDPC Codes*, Univ. Arizona, Tucson, AZ, USA, 2004.
- [2] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo-codes.1," in *Proc. IEEE Int. Conf. Commun.*, vol. 2, 1993, pp. 1064–1070.
- [3] S. J. Johnson, *Iterative Error Correction: Turbo, Low-Density Parity-Check and Repeat-Accumulate Codes*. Cambridge, U.K.: Cambridge Univ. Press, 2010.
- [4] I. Tal and A. Vardy, "How to construct polar codes," *IEEE Trans. Inf. Theory*, vol. 59, no. 10, pp. 6562–6582, Oct. 2013.
- [5] J. Wolf, "Efficient maximum likelihood decoding of linear block codes using a trellis," *IEEE Trans. Inf. Theory*, vol. 24, no. 1, pp. 76–80, Jan. 1978.
- [6] M. P. Fossorier, M. Mihaljevic, and H. Imai, "Reduced complexity iterative decoding of low-density parity check codes based on belief propagation," *IEEE Trans. Commun.*, vol. 47, no. 5, pp. 673–680, May 1999.
- [7] H. Sankar and K. R. Narayanan, "Memory-efficient sum-product decoding of LDPC codes," *IEEE Trans. Commun.*, vol. 52, no. 8, pp. 1225–1230, Aug. 2004.
- [8] E. Rezaei, J. P. Fonseka, and Y. Bo, "Throughput enhancing concatenated codes," *IET Commun.*, vol. 13, no. 9, pp. 1278–1286, 2019.
- [9] E. Rezaei and J. P. Fonseka, "Throughput enhancing concatenated codes with a second uncoded implicit stream," *Electron. Lett.*, vol. 56, no. 1, pp. 32–35, 2020.
- [10] M. Wen et al., "A survey on spatial modulation in emerging wireless systems: Research progresses and applications," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 9, pp. 1949–1972, Sep. 2019.
- [11] T. Mao, Q. Wang, Z. Wang, and S. Chen, "Novel index modulation techniques: A survey," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 1, pp. 315–348, 1st Quart., 2019.

- [12] E. Basar, "Index modulation techniques for 5G wireless networks," *IEEE Commun. Mag.*, vol. 54, no. 7, pp. 168–175, Jul. 2016.
- [13] R. A. Hassan and J. P. Fonseka, "Rate and performance enhancement of LDPC codes using collection of punctured codes decoding (CPCD)," *Int. J. Sens. Wireless Commun. Control*, vol. 11, no. 9, pp. 910–920, 2021.
- [14] R. A. Hassan and J. P. Fonseka, "Improving LDPC and turbo LDPC codes using collection of punctured codes decoding (CPCD)," *Phys. Commun.*, vol. 53, Aug. 2022, Art. no. 101689.
- [15] M. Jiang, C. Zhao, Z. Shi, and Y. Chen, "An improvement on the modified weighted bit flipping decoding algorithm for LDPC codes," *IEEE Commun. Lett.*, vol. 9, no. 9, pp. 814–816, Sep. 2005.
- [16] J. Zhao, F. Zarkeshvari, and A. H. Banihashemi, "On implementation of min-sum algorithm and its modifications for decoding low-density parity-check (LDPC) codes," *IEEE Trans. Commun.*, vol. 53, no. 4, pp. 549–554, Apr. 2005.
- [17] M. Xu, J. Wu, and M. Zhang, "A modified offset min-sum decoding algorithm for LDPC codes," in *Proc. 3rd Int. Conf. Comput. Sci. Inf. Technol.*, vol. 3, 2010, pp. 19–22.
- [18] *NR: Multiplexing and Channel Coding*, 3GPP Standard TS 38.212, 2018.
- [19] *IEEE Standard for Information Technology—Local and Metropolitan Area Networks—Specific Requirements—Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 5: Enhancements for Higher Throughput*, IEEE Standard 802.11n-2009 (Amendment to IEEE Std 802.11-2007 as amended by IEEE Std 802.11k-2008, IEEE Std 802.11r-2008, IEEE Std 802.11y-2008, IEEE Std 802.11w-2009), 2009.
- [20] G. Singh, "A study of encryption algorithms (RSA, DES, 3DES and AES) for information security," *Int. J. Comput. Appl.*, vol. 67, no. 19, pp. 33–38, 2013.



RANA A. HASSAN received the B.Sc. degree in electrical engineering from Cairo University, Egypt, in 2014, the M.Sc. degree in wireless communications from Nile University, Egypt, in 2017, and the Ph.D. degree from the Electrical Engineering Department, University of Texas at Dallas, Dallas, TX, USA, in 2022. She was a Research Assistant with the Wireless Intelligent Network Center, Nile University from 2014 to 2017. Her research interests include wireless communications, channel coding, implicit transmission, proactive networks, and relay networks.



JOHN P. FONSEKA was born in Negombo, Sri Lanka. He received the B.S. degree from the University of Moratuwa, Sri Lanka, the M.S. degree from the Memorial University of Newfoundland, Canada, and the Ph.D. degree in electrical engineering from Arizona State University, USA. He joined as an Assistant Professor of Electrical Engineering with The University of Texas at Dallas in 1988, and is currently serving as a Professor of Electrical Engineering. He currently conducts research in the general area of modulation and coding. Specifically, he is interested in searching for attractive combined modulation and coding techniques, throughput and rate enhancement techniques, efficient decoding techniques, and implicit transmission techniques. He is also interested in the applications of machine learning in communications, implicit transmission, proactive networks, and relay networks.