applied sciences

MDPI

Article

Faster-than-Nyquist Signal Processing Based on Unequal Error Probability for High-Throughput Wireless Communications

Chang-Uk Baek and Ji-Won Jung *

Department of Radio Communication Engineering, Korea Maritime and Ocean University, 727 Taejong-ro, Yeongdo-gu, Busan 49112, Korea; cubaek@kmou.ac.kr

* Correspondence: jwjung@kmou.ac.kr; Tel.: +82-51-410-4424

Received: 22 April 2019; Accepted: 10 June 2019; Published: 13 June 2019

Abstract: Faster-than-Nyquist (FTN) signal processing, which transmits signals faster than the Nyquist rate, is a representative method for improving throughput efficiency sacrificed performance degradation due to inter-symbol interference. To overcome this problem, this paper proposed FTN signal processing based on the unequal error probability to improve performance. The unequal error probability method divides encoded bits into groups according to priority, and FTN interference rates are differently applied to each group. A lower FTN interference ratio is allocated to the group to which high-priority encoded bits belong and a higher FTN interference ratio is allocated to the group to which low-priority encoded bits belong, thus performance improvement can be obtained compared to the conventional FTN method, with the same interference ratio. In addition, we applied the proposed FTN signal processing, based on the unequal error probability method, to the OFDM (orthogonal frequency division multiplexing) system in multipath channel environments. In the simulations, the performance of the proposed method was better than that of the conventional FTN method by about 0.2 dB to 0.3 dB, with an interference ratio of 20%, 30%, and 40%. In addition, in multipath channels, we confirmed that by applying the proposed unequal error probability, the OFDM-FTN method improves performance to a larger extent than the conventional OFDM-FTN method.

Keywords: FTN; inter-symbol interference; unequal error probability; priority; OFDM; OFDM-FTN

1. Introduction

Next generation wireless and/or mobile communications require a high transmission efficiency and high reliability to provide various services to subscribers. Recently, many methods for increasing the throughput are being researched, as the demand for next-satellite broadcasting communication and 5G (5th Generation Mobile Telecommunication)-based mobile communication for throughput increases, while the bandwidth is limited [1,2]. Research on applying the multidimensional modulation method in the modulator/demodulator and speeding up the high-speed channel encoder/decoder to meet the high data rate is already saturated. Thus, the research direction has focused more on transmission methods. However, it is very difficult to improve both throughput and performance, because the two are in a trade-off relationship. Therefore, it is highly important to develop methods that can maintain the maximum performance, while increasing the throughput.

Recently, the faster-than-Nyquist (FTN) method [3–5], which transmits faster than the throughput of Nyquist, is emerging as the standard for the next generation Digital Video Broadcasting-Satellite-Third Generation (DVB-S3) and the future of broadcast television (FOBTV) [6,7]. It is possible to transmit signals faster than the Nyquist rate and maximize throughput with the same channel bandwidth. However, the FTN method has a limitation in maximizing the transmission efficiency due to the performance degradation arising from inter-symbol interference (ISI).

The multiple input/multiple output-FTN (MIMO-FTN) [8–10] transmission method, which combines the MIMO and FTN methods to improve throughput, can maximize the improvement of the throughput, although its method of decoding and removing interference is difficult, the research about which is still underdeveloped.

In [8], space–time trellis coding and the weighted–zero forcing algorithm were applied to distinguish the symbols of each transmit antenna, and the Bahl, Cocke, Jelinek, and Raviv (BCJR) algorithm [11,12] was used to cancel the interference in order to improve the error performance by increasing the number of iterations. However, it did not improve the performance compared to the single input/single output-FTN (SISO-FTN) method.

In the paper, we employed the low-density parity-check (LDPC) coding technique, as a channel coding scheme, which is defined based on a sparse parity-check matrix, H, where N bit nodes and $(\mathrm{N}-\mathrm{K})$ check nodes are connected by edges [13,14]. In the H matrix, the bit nodes/check nodes are mapped to the columns/rows, where one element in a row is connected to another in a column by an edge. Bit nodes consist of information and parity bit nodes, while information bit nodes are randomly connected by edges with check nodes by features of the matrix, H. To improve the bit error rate (BER) performance of the FTN signal with LDPC codes, we proposed FTN signal processing based on the unequal error probability (UEP) method in the SISO channel. The UEP method divides encoded bits into groups, according to their priority and FTN interference, which are differently applied to each group. In deciding which group has a higher priority, the horizontal shuffle schedule (HSS) LDPC decoding algorithm is employed. The HSS-based LDPC decoding algorithm, proposed in [15,16], is used to accelerate the LDPC decoding speed. However, in this study, it was used to configure the priority algorithm. The LDPC decoder performs an iterative exchange between the bit nodes updating and the check node updating until the maximum number of iterations has been reached. Conventional LDPC decoder calculates bit nodes and check nodes updating sequentially based only on messages updated in the previous iteration. However, in the HSS-based LDPC decoding process, the already updated bit node and check node in the same iteration move to a different bit and check node. Therefore, bit node groups connected to check nodes, which are processed first, are assigned a higher priority. By classifying the groups according to the importance of the encoded bits, based on the decoding process, the proposed algorithm improves the performance in the same overall interference ratio compared to that of the conventional FTN method.

Lastly, we applied the proposed UEP-FTN method to an orthogonal frequency division multiplexing (OFDM) system in multipath channels. Through simulation, the performance of the conventional FTN method and FTN based on the UEP transmission method were compared. Additionally, the performance of the OFDM-FTN transmission method applying UEP was analyzed in a multipath channel environment.

2. System Model of FTN Signal Processing

2.1. FTN Signal Processing

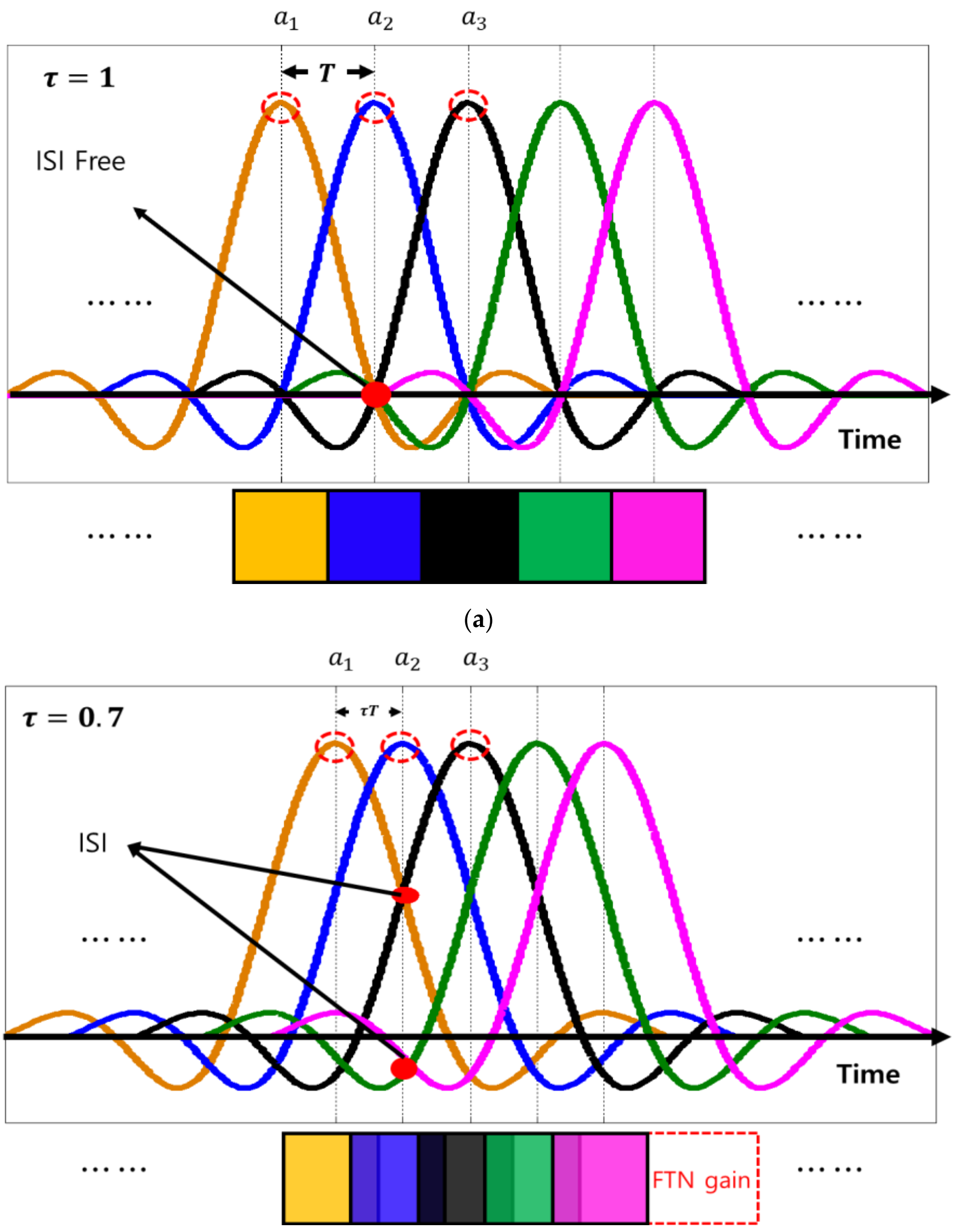
FTN signaling is a technique for transmitting information at a rate higher than the allowed Nyquist limit in order to improve the transmission efficiency by applying the FTN method, which increases the interference rate. However, FTN signaling has a limitation in maximizing the transmission efficiency due to performance degradation arising from ISI. The transmission signal with ISI can be given by Equation (1):

$$\mathrm{s(t)} = \sqrt{E_s}\sum\nolimits_n a_n h(t-n\tau T),\ \tau<1, \tag{1}$$

where a_n is the encoded bit stream; E_s is the average symbol energy; h(t) is a unit-energy baseband pulse, which, for this paper, we will assume is orthogonal to shifts by T; and τ is the interference time. The interference ratio, τ′, is given by:

$$\tau'(\%) = 100\times(1-\tau). \tag{2}$$

Figure 1a shows that no ISI is generated, as the transmission is run at the Nyquist rate, when $\tau = 1$. However, in Figure 1b, $\tau = 0.7$, the adjacent symbols affect each other due to FTN, and, as a result at each decision point of data, the parts labeled with a red circle are added to the raw data and affect them. Thus, we know that there is a change in the waveform due to the interference. While the signal's waveform becomes distorted due to ISI, if this issue is overcome, it can be seen that the transmission rate improves by 30% at the same time.



(**a**)

(**b**)

Figure 1. FTN signal (roll off factor = 0): (**a**) $\tau = 1$; (**b**) $\tau = 0.7$.

2.2. *Conventional Transceiver Model for FTN Signal Processing*

Figure 2 shows the transceiver model based on FTN signal processing.

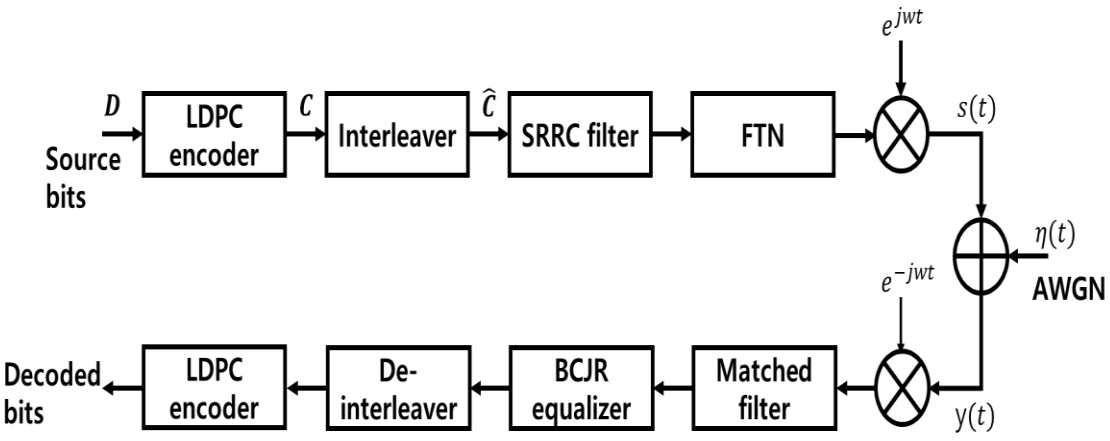


Figure 2. The structure of the FTN model.

The source bits to be transmitted by bit-stream, ***D***, are given by:

$$\boldsymbol{D} = \{d_1, d_2, \cdots, d_K\}, \tag{3}$$

where ***K*** is the size of ***D***. First, ***D*** is encoded by the (N, K) LDPC encoder. The coded bit stream, ***C***, is given by:

$$\boldsymbol{C} = \{\boldsymbol{c_1}, \boldsymbol{c_2}, \cdots, \boldsymbol{c_N}\}, \tag{4}$$

where ***N*** is the size of the coded bit. The encoded bits passed through the encoder transform the burst error into the random error via the interleaver. The interleaved output is canceled a posteriori from the proceeding received signal. The interleaving function helps the receiver convergence. $\hat{\boldsymbol{C}}$ means the encoded bits have passed through the interleaver:

$$\hat{\boldsymbol{C}} = \{\hat{\boldsymbol{c}}_1,\ \hat{\boldsymbol{c}}_2, \cdots, \hat{\boldsymbol{c}}_N\}. \tag{5}$$

FTN signaling is a technique for transmitting information at a rate higher than the allowed Nyquist limit. Consequently, ISI necessarily occurs.

The interference transmission signal, $s(t)$, is given by:

$$\boldsymbol{s(t)} = \sum\nolimits_{\boldsymbol{n=1}}^{\boldsymbol{N}} \hat{\boldsymbol{C}}_{\boldsymbol{n}} \boldsymbol{h(t-n\tau T), \tau < 1,} \tag{6}$$

where $\hat{C}_n$ is the encoded bit stream, after interleaving; $h(t - n\tau T)$ is a unit-energy baseband pulse; τ is the interference time; and T is the symbol duration.

The received signal, after the demodulation process and the matched filter operation, are inputted into the BCJR equalizer to remove ISI. The BCJR equalizer removes an amount of FTN interference by a trellis diagram, which consists of a branch, backward, and forward metric. The number of decoding trellis states of the BCJR is 32 [11,12]. The output values of the BCJR equalizer are inputted into the LDPC decoder, after passing through the de-interleaver.

3. FTN Signal Processing Based on the UEP Algorithm

The UEP-based FTN transmission method applies the FTN interference ratio of transmission signals, based on the importance of the encoded bits, obtained using the priority algorithm. This algorithm can determine the importance of each transmitted bit using the HSS-based LDPC decoding algorithm. The HSS-based LDPC decoding algorithm, proposed in [15,16], is used to accelerate the LDPC decoding speed. However, in this study, it was used to configure the priority algorithm.

Figure 3 shows the structure of the transceiver for the FTN based on the UEP algorithm.

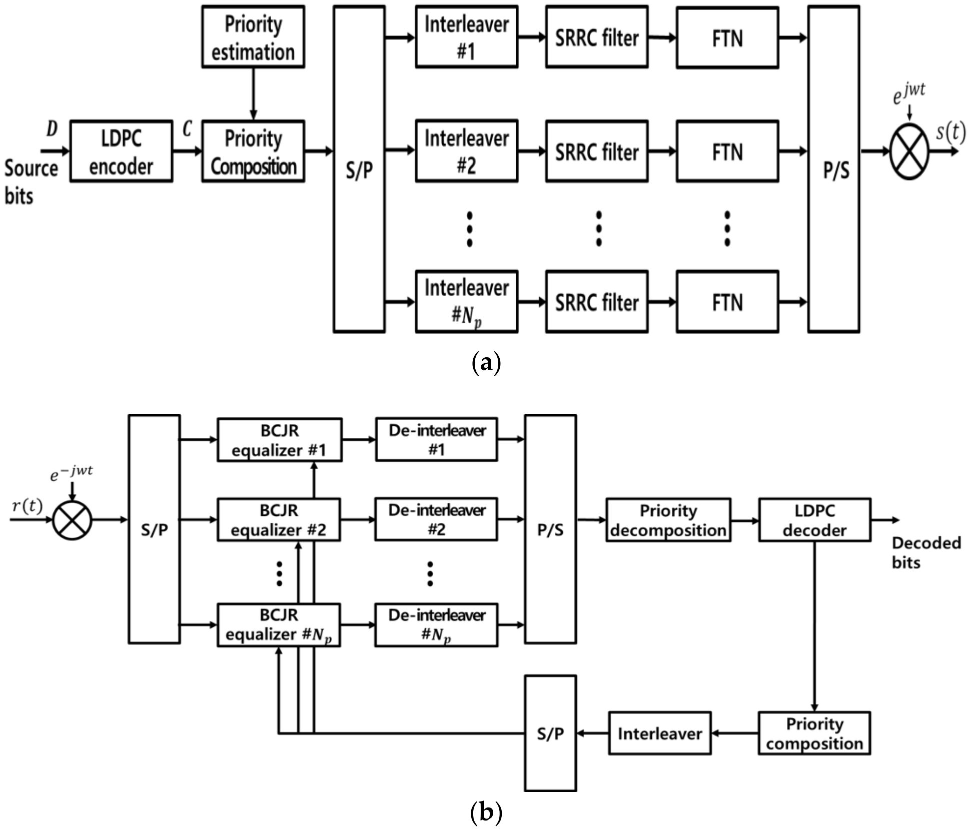


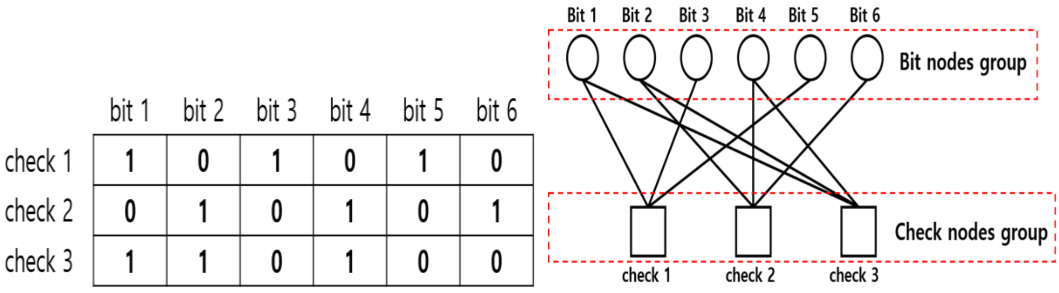
Figure 3. The transceiver structure of the FTN model based on the UEP algorithm: (**a**) Transmitter; (**b**) receiver.

As shown in Figure 3, the FTN signal processing-based UEP method first performs the LDPC encoding process using source bits. For the UEP setting, the priority algorithm is used to set up the priority of the codes or signals being transmitted, and bit nodes are classified preferentially using HSS-based LDPC decoding at the receiving section. In this process, the H matrices of the LDPC encoder are utilized. After being prioritized and arranged through priority blocks, codes are arranged in groups through the serial to the parallel (S/P), according to the number of interference ratios to be used. After passing through the interleaver, a transmitting signal is generated by setting the FTN interference rate in each group differently.

Transmit signals that pass through the additive white Gaussian noise (AWGN) channels go through a demodulation process. Moreover, an amount of FTN interference is removed through the BCJR equalizer. After de-interleaving, reordered signals arranged by the UEP algorithm are inputted into the LDPC decoder. The iterative turbo equalizer consists of inner codes and outer codes, and we the BCJR is employed as an outer code and the LDPC code as an inner code. At the receiver side, powerful turbo equalization algorithms that iteratively exchange probabilistic information between the inner decoder and outer decoder are used, thereby reducing the error rates significantly. The inner coded bits are then subtracted from the input and interleaved. The interleaved output is canceled a posteriori from the proceeding received signal. Thus, the algorithm enhances the performance, while maintaining or improving the transmission efficiency of the existing FTN transmission method.

3.1. UEP Setting Algorithm for the HSS-Based LDPC Decoding Method

For the UEP-based FTN signals to be generated, it is necessary to divide encoded bits using the priority algorithm and to allot different FTN interference ratios accordingly. In this process, the priority algorithm determines the importance of each bit. In this section, we explain the HSS-based LDPC decoding algorithm briefly and how to apply it to the UEP setting algorithm. The (N, K) LDPC code is defined based on a sparse parity-check matrix, H, where N bit nodes and (N – K) check nodes are connected by edges. In the H matrix, the bit nodes/check nodes are mapped to the columns/rows, where one element in a row is connected to another in a column by an edge. Bit nodes consist of information and parity bit nodes, while information bit nodes are connected by edges with check nodes by features of the matrix, H. The LDPC codes are shown in the small example (6, 3), shown in Figure 4.



	bit 1	bit 2	bit 3	bit 4	bit 5	bit 6
check 1	1	0	1	0	1	0
check 2	0	1	0	1	0	1
check 3	1	1	0	1	0	0

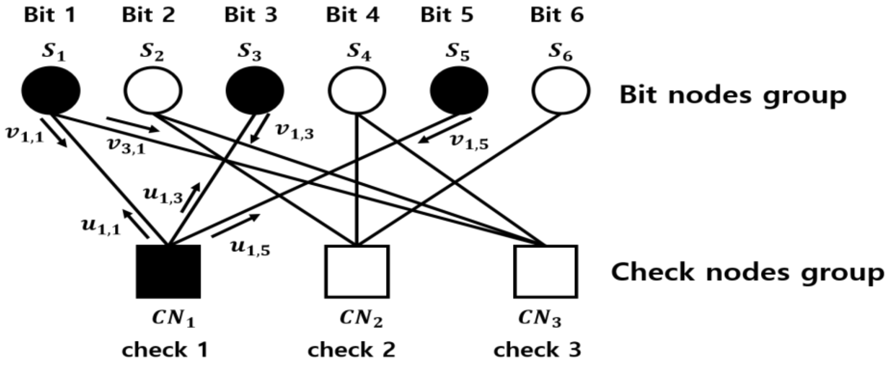
(**a**) (**b**)

Figure 4. Representation of an LDPC code using (**a**) a parity-check matrix **H** and (**b**) Tanner graph.

Figure 4b is a Tanner graph, which can be deduced from the parity-check matrix of Figure 4a. In Figure 4b, each bit is represented by a bit node (shown as a circle), and each check is represented by a check node (shown as a square). An edge exists between the bit node, $\boldsymbol{j}$, and the check node, $\boldsymbol{i}$, and only if $\boldsymbol{H(i,j) = 1}$. The LDPC decoder performs an iterative exchange between the bit nodes and the check nodes, until the maximum number of iterations has been reached. The decoding algorithm produces the probabilities of each bit being "0" or "1" for every iteration. These probabilities are used in the log likelihood ratio (LLR) and are computed in the following way:

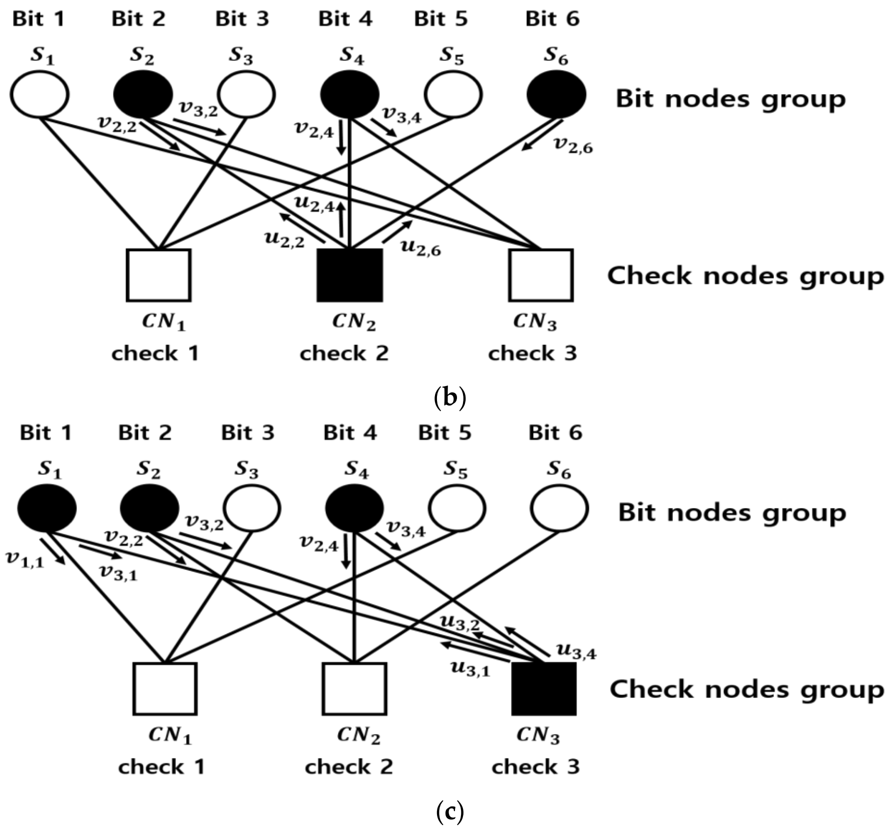
$$\boldsymbol{LLR_i(bit_i) = log\left(\frac{P(bit_i = 0)}{P(bit_i = 1)}\right)}. \tag{7}$$

The Tanner graph of the (6, 3) LDPC code, as shown in Figure 5, is an example to explain the HSS decoding process.



(**a**)

Figure 5. *Cont.*



(**b**)

(**c**)

Figure 5. Tanner graph and the HSS-based LDPC decoding process: (**a**) Step 1: first check of the node update; (**b**) Step 2: second check of the node update; (**c**) Step 3: third check of the node update.

Each bit node updates all edges of $u_{i,j}$, which are linked from the i^{th} check node to the j^{th} bit node, S_j. The final decision on a bit is taken at the end of the decoding process by computing the sign of the local a posteriori log density ratio:

$$S_j = LLR_j + \sum\nolimits_{k=1}^{d_v} u_{i,j}, \tag{8}$$

where d_v means the number of edges linked to the j^{th} bit node. In Equation (8), as shown in Figure 5a, if j is 1, i is equal to 1 and 3. Depending on whether S_j is positive or negative, the bit, j, is considered as a "0" or "1". As shown in Figure 5, the process of updating a check node and bit node is defined from Step 1 to Step 3. Computing the probability of edge, $u_{i,j}$, which is linked from the i^{th} check node to the j^{th} bit node for a degree, d_c, is as follows Equation (9):

$$u_{i,j} = \oplus_{k=1,n\neq j}^{d_c} v_{i,n}, n \in CN_i, \tag{9}$$

where d_c denotes the number of edges linked to the check node. $v_{i,n}$ is the probability of the edge, which is linked from the j^{th} bit node to the i^{th} check node, and n is the index of the bit node group, which is linked to the check node, CN_i. The ⊕ operator is defined as follows:

$$v_i \oplus v_j = sign(v_i) \times sign\left(v_j\right) \times (\min\left(|v_i|, \left|v_j\right|\right) - \text{offset}, \tag{10}$$

$$\text{offset} = \log\left(\frac{1 + e^{-(|v_i| + |v_j|)}}{1 + e^{-(|v_i| - |v_j|)}}\right). \tag{11}$$

The edge, $v_{i,j}$, is updated for a degree, d_v, which can be expressed as follows (Equation (12)):

$$v_{i,j} = LLR_j + \sum_{k=1, n \neq i}^{d_v} u_{n,j}, n \in S_j, \tag{12}$$

where d_v denotes the number of edges linked to the j^{th} bit node. n is the index of the check node group, which is linked to the bit node, S_j.

In the conventional LDPC decoding process, the $u_{i,j}$ coming out of the check node is updated using Equations (9)–(11). Then, all $v_{i,j}$ messages coming out of the bit nodes linked to this check node are updated with Equation (12). In the l^{th} iteration, the update of the bit node is based only on messages updated in the $(l-1)^{th}$ iteration. However, in the HSS-based LDPC decoding process, the already updated bit node and check node in the l^{th} iteration move to a different bit and check node. As an example, in Figure 5, $v_{3,1}$ was already updated two times, before being processed by the third check node. This higher update frequency per iteration allows the decoder to converge faster, because there are fewer iterations.

The HSS-based LDPC decoding algorithm, proposed in [15,16], is used to accelerate the LDPC decoding speed, because it needs fewer iterations to maintain the same performance. However, in this study, it was used to configure the priority algorithm. In deciding which group has the higher priority, the HSS-based LDPC decoding algorithm, which is explained above, was applied. Unlike conventional LDPC decoding processing, which calculates bit nodes' and check nodes' updating sequentially based only on messages updated in the previous iteration, in the HSS-based LDPC decoding process, the already updated bit node and check node in the same iteration move to a different bit and check node. Therefore, the bit node groups linked to the check nodes, which are arranged on the left, are assigned a higher priority. For example, as shown in Figure 5, the bit node group, which is linked to the first check node in Step 1, is the most important bit group. By classifying the group depending on the importance of the encoded bits, based on the decoding process, the UEP algorithm applies different FTN interference ratios according to the importance of the encoded bits, and improves the performance in the same overall interference to a higher degree than the conventional FTN method.

Figure 6 shows the arrangement of the bit node and check node groups for the priority algorithm of the UEP setup. N_{G_c} and N_{G_b} mean the number of check node groups and bit node groups, respectively.

In the half rate LDPC codes in the Digital Video Broadcasting-Satellite-Second Generation (DVB-S2) standard with N = 64,800, the number of check nodes is 32,400 and one check node group has 360 data, therefore, N_{G_c} = 90. Additionally, the number of bit nodes is 64,800 and one bit node group has 360 data, therefore, N_{G_b} = 180. In Figure 6, the left part shows the sequence of bit nodes, transmitted in the conventional way. For instance, without the UEP setting, signals are transmitted in the order from the 1st bit node group to the 180th bit node group (see the left part of Figure 6). However, we rearranged the bit node group linked to the check node group, which was calculated in advance (see the right part of Figure 6). For example, the 26, 27, 30, 36, 74, 90, and 179th bit node groups, linked to the first check node group, should have a high priority. The priority algorithm set the priority of the bit node group, linked with the 1st to 90th check node group. Hence, as shown in the right part of Figure 6, it is necessary to rearrange the bit nodes to be transmitted firstly according to the HSS-based LDPC decoding process of the priority algorithm.

For instance, with no UEP setting, signals are transmitted in the order from the 1st bit node group to the 180th bit node group. If the priority algorithm configures and sends the UEP setting, the 26th bit node group, connected to the 1st check node group, is transmitted instead of the 1st bit node group, as shown in the right part of Figure 6. Thus, the priority algorithm for the UEP setting determines the priority of groups and arranges them according to the positions of the bit node groups connected to the check node groups in the H matrix of the code rates to be transmitted.

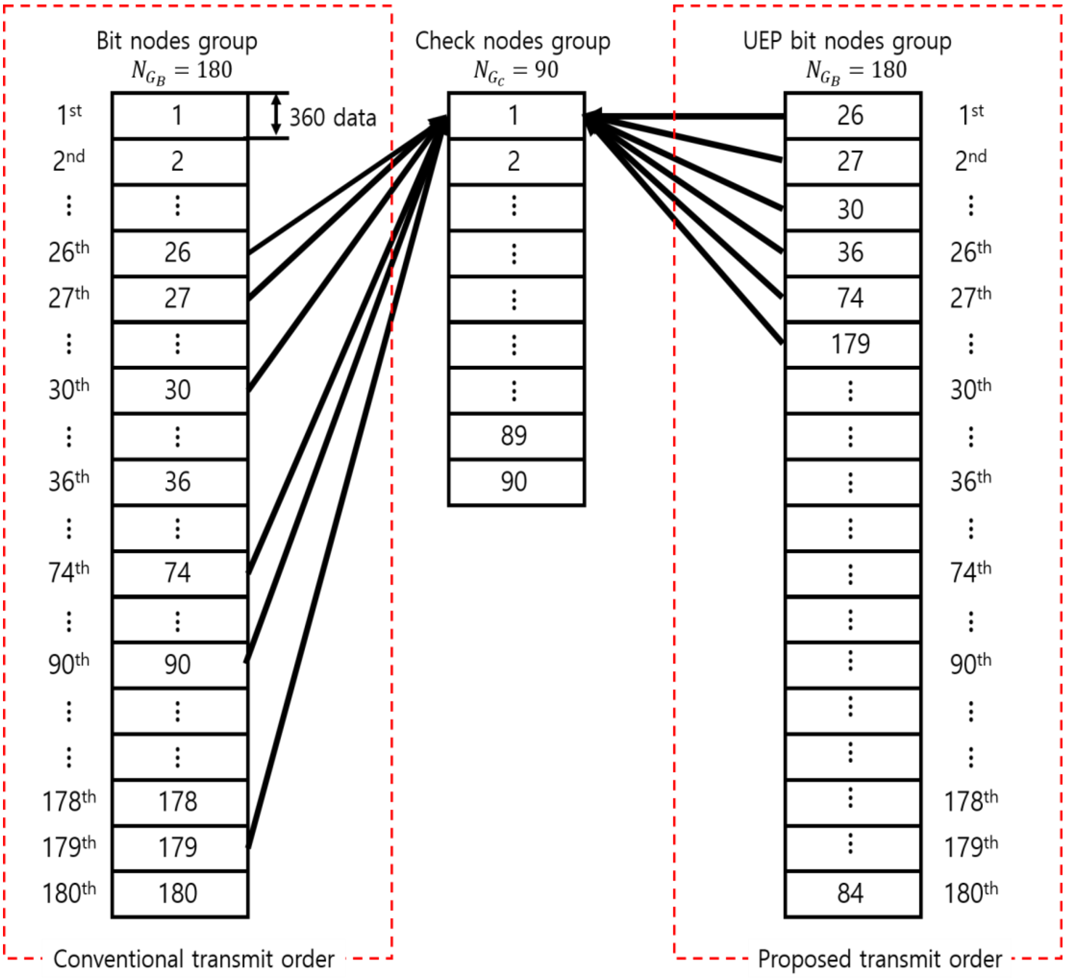


Figure 6. The order of the bit node group, with or without the UEP algorithm.

3.2. Reordering According to Priority Settings

Figure 7 shows the composition of the encoded bits, according to their priority. As explained in Section 3.1, we reordered the LDPC-coded bits, from high priority to low priority, from the right part of Figure 6 to the left part. For example, the 26th bit node group of the 360 coded bits were firstly rearranged in the form of Figure 7.

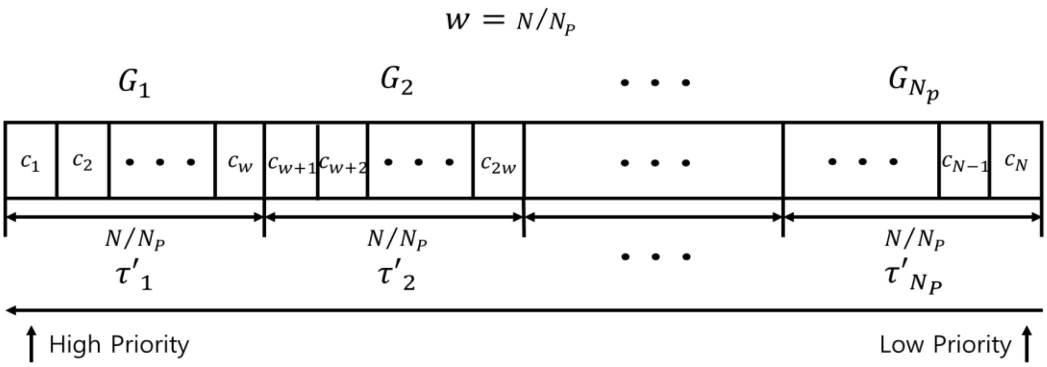


Figure 7. Composition of the code word by priority.

$\boldsymbol{N_p}$ is the number of different FTN interference rates, used to perform the UEP setups by applying a priority algorithm for the $\boldsymbol{G_{N_p}}$ groups. For example, N = 64,800 LDPC coded bits, supposing $\boldsymbol{N_p = 4}$, and each group, $\boldsymbol{G_i}$, has 16,200 coded data.

For high-priority bit groups, the FTN interference ratio needs to be set to low, as this can enhance the performance, because the number of repetitions between the bit node and check node updating processes is high during decoding. For low-priority bit groups, the FTN interference ratio needs to be set to high, because the number of bit node and check node updating repetitions is relatively small. High and low priorities indicate the level of importance of the transmitted signals, and each group of transmitted signals have different FTN interference ratios, based on the priority. A transmitting signal is generated by setting the FTN interference rate in each group differently. The average of $\tau'_1, \tau'_2, \cdots, \tau'_{N_p}\left(\tau'_1 \le \tau'_2 \le \cdots \le \tau'_{N_p}\right)$ for different interference ratios is set to be the same as that in the conventional FTN transmission methods. Thus, the algorithm enhances the performance, while maintaining or improving the transmission efficiency of the conventional FTN transmission method.

3.3. Simulation Results

The performance of the proposed FTN-based UEP method and that of the conventional FTN method were compared through simulation. The simulation was performed using the Visual C++ language on a PC. The LDPC encoder employed K = 32,400 as the size of the source bit, N = 64,800 as the size of coded bits, and R = 1/2 as the rate of encoding. The square root raised cosine (SRRC) filter, with a roll-off factor of 0.35 as defined by the DVB-S2 standard, was used. The performance was compared by setting the FTN interference rate to 20%, 30%, and 40.0%.

The UEP setting was configured based on the number of interference ratios, $N_p = 4$. Fixed on the LDPC coding parameters, shown in Table 1, in order to achieve the performance improvement effect of the UEP algorithm, the different interference ratios for the UEP setting were configured, and the average interference ratios were set to 20%, 30%, or 40%. The UEP-based FTN transmission method was applied. To compare the conventional FTN algorithm, with the same interference ratio applied to all groups, we performed a BER performance analysis according to the different interference ratios allotted for the priority group, as shown in Table 2.

Table 1. Simulation parameters.

Total number of data	10^6
Channel coding	LDPC (HSS algorithm)
Input bit (K)	32,400
Code word (N)	64,800
Coding rate	1/2
Modulation	BPSK
Number of sampling bits (SRRC filter)	32
Roll off factor	0.35
Interference rate (τ')	20%, 30%, 40%
LDPC inner iteration (l)	60
N_p	4

Table 2. UEP parameter settings for the average of the interference ratio, $\tau' = 20\%$.

	Interference Rate for Each Priority Group (%)			
	τ'_1	τ'_2	τ'_3	τ'_4
Type1	22	21	19	18
Type2	18	19	21	22
Type3	30	25	15	10
Type4	10	15	25	30

To investigate which cases are more efficient for the UEP-FTN algorithm, various cases were compared in a simulation: Large interference ratio differences were assigned between groups, and groups were assigned either high or low interference ratios.

The average of the interference ratio is set to 20% in Table 2. Type 1 and type 2 are the cases in which the difference in the interference ratio between each group was set to low. Type 3 and type 4 are the cases in which the difference in the interference ratio between each group was set to high. In addition, type 1 and type 3 are cases in which the high interference ratios were set in the high-priority group. Type 2 and type 4 are cases in which the high interference ratios were set in the low-priority group. The BER performance of the four types is shown in Figure 8.

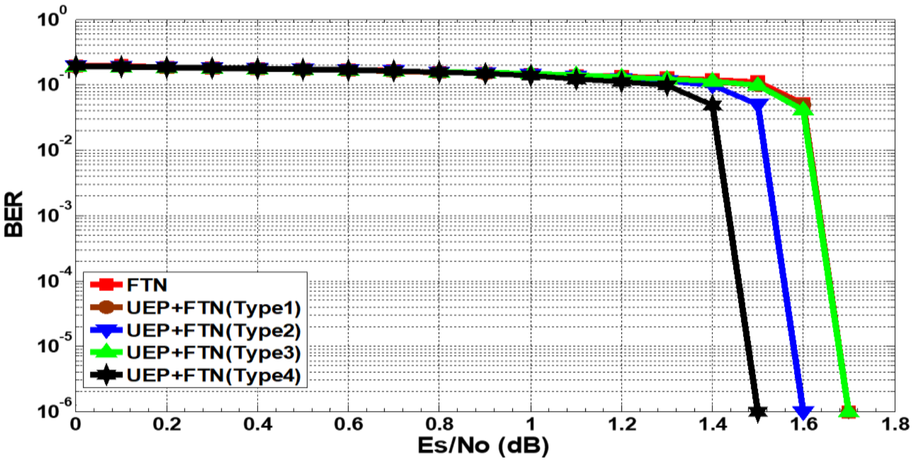


Figure 8. The performance of the FTN method based on the UEP algorithm ($\tau' = 20\%$).

The simulation result shows that the performance was improved in type 4 and type 2, where the range of the interference ratio was set to high, and a low interference ratio was applied to the high-priority group. Type 4 has the best performance compared to the other types. Furthermore, when a high interference ratio was applied to the high-priority group, as in type 1 and type 3, the performance was the same as that of the conventional FTN. Thus, we confirmed that the low interference ratio should be applied to the high-priority group, as in type 4, and the difference in the interference ratios between groups needs to be large.

Based on type 2 and type 4, a simulation was conducted, with the interference of the FTN transmission method set to 30% and 40%. The interference ratio of each priority group was set, as in Tables 3 and 4.

Table 3. UEP parameter settings for the average of the interference ratio, $\tau' = 30\%$.

	Interference Rate for Each Priority Group (%)			
	τ_1'	τ_2'	τ_3'	τ_4'
Type2	27	29	31	33
Type4	15	20	40	45

Table 4. UEP parameter settings for the average of the interference ratio, $\tau' = 40\%$.

	Interference Rate for Each Priority Group (%)			
	τ_1'	τ_2'	τ_3'	τ_4'
Type2	38	39	41	42
Type4	30	35	45	50

The performance applied in Tables 3 and 4 is shown in Figures 9 and 10.

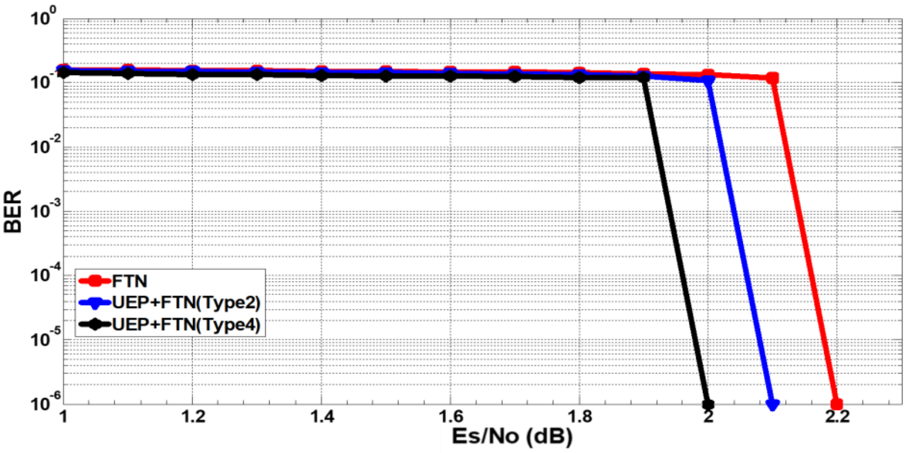


Figure 9. The performance of the FTN method based on the UEP algorithm ($\tau' = 30\%$).

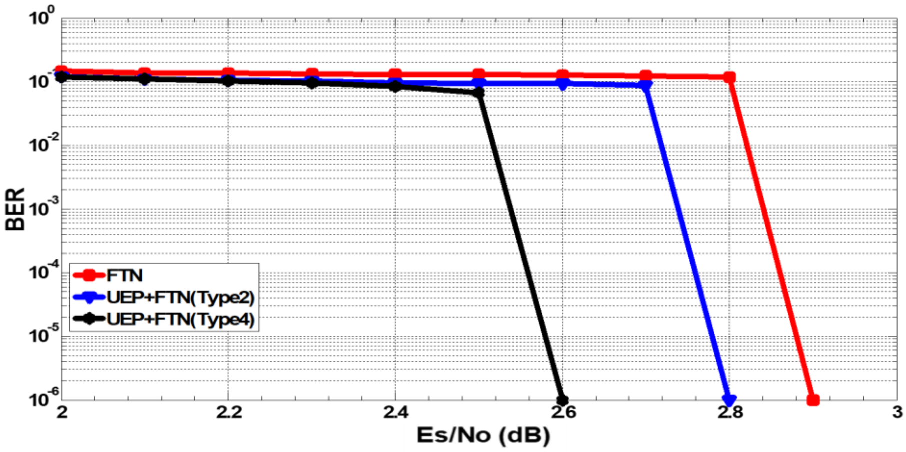


Figure 10. The performance of the FTN method based on the UEP algorithm ($\tau' = 40\%$).

In Figures 9 and 10, the red BER curves represent the performance, with the interference ratio set at 30% and 40% in the FTN, without UEP. Compared to FTN without UEP at the BER of 10^{-4}, it was demonstrated, in the simulation, that the performance improved by 0.1 dB at BER = 10^{-4} in type 2, where the range of the FTN interference ratio between groups was set to low in the UEP-FTN algorithm. In addition, it was shown that the performance improved by 0.2 dB and 0.3 dB in the case of an average interference ratio of 30% and 40%, respectively, in type 4, where the range of the FTN interference ratios between groups was set to high. The simulation demonstrated that the most excellent performance was obtained in type 4, where the range of the interference ratios for each group was set to high through each priority algorithm, and the low interference ratio was applied to the high-priority group.

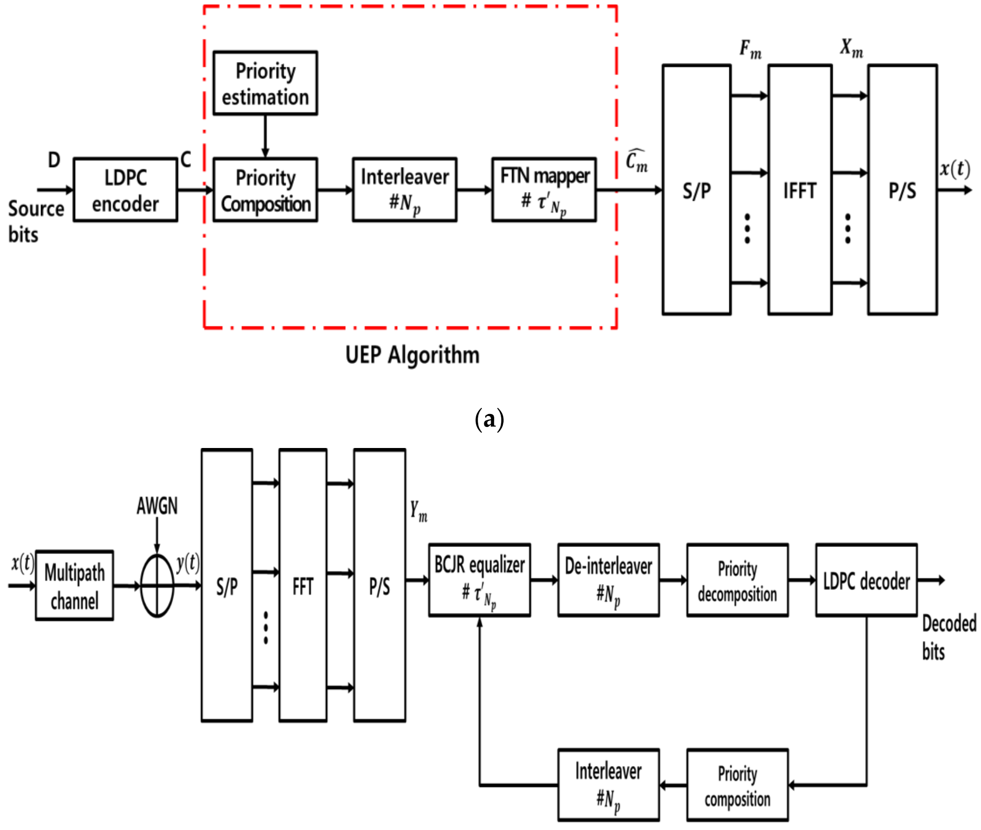
These results can be interpreted as follows. First, the reason why the range of the interference ratios was set to high was because the difference in the branch metric allocated to each trellis became larger, allowing the BCJR decoder to operate efficiently. Second, applying the low interference ratios was applied to the high-priority signal in order to enhance the reliability of the transmitted signal, the LDPC decoder can update the highly reliable data by the calculation of the bit node and check node updating. Through this process, the performance can eventually be improved.

The simulation result shows that the UEP-based FTN transmission method, with the applied priority algorithm, demonstrated a better performance than the conventional FTN transmission method.

4. OFDM-FTN Applications Based on the UEP Algorithm

The OFDM method [17,18] divides data into multiple carrier signals and multiplexes and transmits them by adding orthogonality to minimize the interval between the divided carrier signals. Since the OFDM with a high spectrum efficiency is efficient in a frequency-selective fading channel environment and a multipath channel environment, it can maximize the band efficiency by applying the FTN method and also guarantee the performance in multipath channels. We proposed OFDM-FTN applications based on the UEP algorithm, which shows an excellent performance in a multipath channel environment and has a high spectrum efficiency using multiple subcarriers.

Figure 11 shows the structure of the proposed OFDM-FTN applications based on the UEP algorithm.



(a)

(b)

Figure 11. The transceiver structure of OFDM-FTN model based on the UEP algorithm: (**a**) Transmitter; (**b**) receiver.

The source bits to be transmitted by the bit-stream, $\boldsymbol{D}$, and the coded bit stream, $\boldsymbol{C}$, passed through the encoder are expressed in Equations (3) and (4), respectively. Through the UEP algorithm, code words are divided into groups by the number of interference ratios. Moreover, each group applies a different interference ratio. The encoded data, applied with the UEP, are divided into the subcarrier number by the S/P and are expressed as $\hat{C}_m$, as shown in the following Equation (13):

$$\hat{C}_m = \left\{\hat{c}_1, \hat{c}_2, \cdots, \hat{c}_{N/M}\right\}. \tag{13}$$

In this equation, $\boldsymbol{M}$ is the number of sub-carriers, $\hat{\boldsymbol{C}}_m$ passes through the inverse fast Fourier transform (IFFT), and the transmitted signal is expressed in the following Equation (14):

$$x(t) = \sum\nolimits_{m=0}^{M-1} X_m(t) e^{j2\pi tm/M}, \tag{14}$$

where $X_m(t)$ are the OFDM symbols of the m-th subcarriers.

The transmission signal is passed through the multipath and the AWGN channel. The received signal is given by:

$$\boldsymbol{y(t)} = \sum\nolimits_{l=0}^{L-1} \boldsymbol{x(t-l)h_l(t)} + \boldsymbol{\eta(t)}, \tag{15}$$

where L denotes the total number of multipaths, means the l-th path of all multipaths, and $h_l(t)$ refers to the channel response coefficient on the l-th path of all multipaths.

The received signal, $y(t)$, is divided into M by S/P and passes through the FFT (fast Fourier transform). After FFT, $Y_m(t)$ is given by:

$$\boldsymbol{Y_m(t)} = \sum\nolimits_{n=0}^{N/M-1} \boldsymbol{y(t)e^{-j2\pi nm/M}}. \tag{16}$$

$Y_m(t)$ are inputted into the BCJR equalizer to remove ISI due to FTN. The output values of the BCJR equalizer are inputted into the LDPC decoder, after passing through the de-interleaver.

Simulation Results

The performance of the OFDM-FTN transmission method, which combines the efficient OFDM method in multipath channels with the FTN method to enhance the transmission ratio, was analyzed through simulations. In addition, the proposed UEP algorithm was applied to the OFDM-FTN transmission method, and the performance was comparatively analyzed. The simulation parameters are the same as those shown in Table 1 in Section 3.3. The number of the interference rate was 4, which is N_p = 4. M was the number of subcarriers. The subcarriers, M, of the OFDM-FTN method were set to 4, 8, 16, and 32. The multipath channel environment was simulated by setting the multipath number, L, to 3 in Equation (13) and setting the channel response coefficients, h(0) to 0.8, h(1) to 0.2, and h(2) to 0.1.

Figure 12 shows the performance of the conventional FTN method and OFDM-FTN method in a multipath channel environment. We compared the performance by setting the FTN interference rate to 30%. The simulation results for the conventional FTN method could not be decoded in the multipath channel environment. However, data for the OFDM-FTN method could be decoded in the multipath channel environment. In addition, we confirmed that the performance improved as the subcarrier number, $\boldsymbol{M}$, of the OFDM-FTN transmission method increased. In order to overcome the degradation of performance due to the interference ratio in the OFDM-FTN transmission method in the multipath channels, the proposed UEP was applied to the OFDM-FTN transmission method. The simulation was performed by applying type 4 in Table 3, and the number of subcarriers, M = 4, 8, 16, and 32.

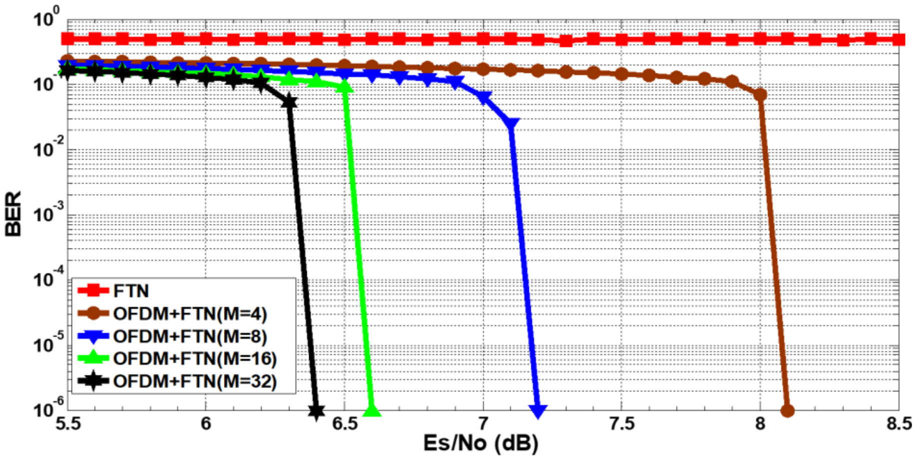


Figure 12. The performance of FTN and OFDM-FTN in the multipath channel ($\tau' = 30\%$).

Figure 13 shows that the number of M subcarriers of the OFDM-FTN method was set to 4, 8, 16, and 32, with UEP settings based on the FTN interference ratio of 30%. The UEP settings shown in type 4 of Table 3 were applied to the simulations. According to Figure 13, the proposed OFDM-FTN based on the UEP method improved the performance by approximately 0.1 to 0.3 dB, compared to the OFDM-FTN method, at the BER of $\mathbf{10^{-4}}$. We confirmed that the OFDM-FTN method applying the UEP method also has a better performance than that of the OFDM-FTN method in multipath channels.

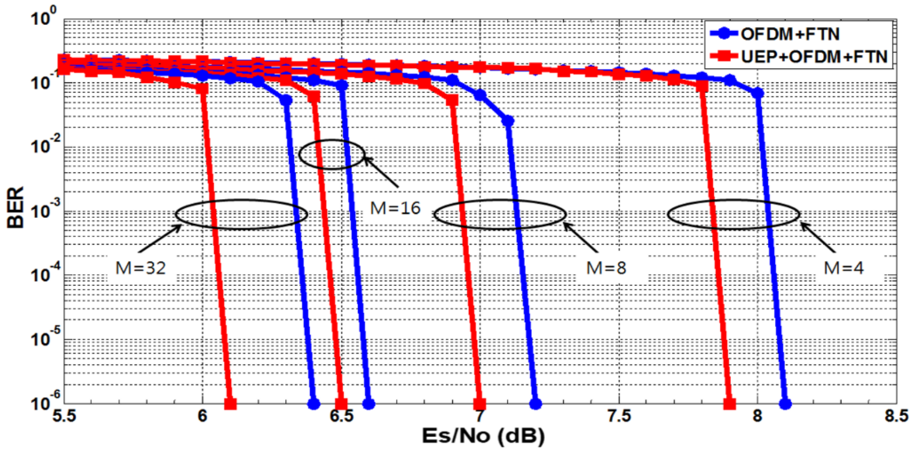


Figure 13. The performance of the OFDM-FTN method based on the UEP algorithm ($\tau' = 30\%$).

5. Conclusions

This paper proposed UEP-based FTN signal processing to compensate for performance degradation due to symbol interference. The UEP method divided encoded bits into groups according to priority, and FTN interference ratios were differentially applied to each group. In deciding which group had the higher priority, the HSS-LDPC decoding algorithm was applied. By employing the LDPC coding method and FTN interference ratios of 20%, 30%, and 40%, we confirmed that the performance of the proposed UEP-FTN method, when applying the larger interference ratio to each group, improved by about 0.2 to 0.3 dB at the BER of $\mathbf{10^{-4}}$, compared with the conventional FTN method.

In addition, we applied the UEP-based FTN algorithm to OFDM applications in a multipath channel, with an interference ratio of 30%. It was confirmed that the data could be decoded in the multipath channel, and the performance could be improved by 0.1 to 0.3 dB, as the number of subcarriers of the OFDM-FTN method increased.

In the paper, we confirmed that the UEP-FTN method has both a performance improvement and high transmission efficiency, and it is useful for next-satellite and/or wireless mobile communication.

Author Contributions: C.-U.B. and J.-W.J. conceived and designed the simulations; C.-U.B. performed the numerical simulations and analyzed the data; C.-U.B. wrote the paper; and J.-W.J. revised the paper and offered useful writing suggestions.

Acknowledgments: The research was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF), funded by the Ministry of Education (**NRF-2017R1D1A1A09082161**).

Conflicts of Interest: The authors declare no conflict of interest.

Data Availability Statement: The data used to support the findings of this study are available from the corresponding author upon request.

References

1. Gupta, A.; Jha, R.K. A survey of 5G network: Architecture and emerging technologies. *IEEE Access* **2015**, *3*, 1206–1232. [CrossRef]
2. Larsson, E.G.; Edfors, O.; Tufvesson, F.; Marzetta, T.L. Massive MIMO for next generation wireless systems. *IEEE Commun. Mag.* **2014**, *52*, 186–195. [CrossRef]
3. Mazo, J.E. Faster than Nyquist signaling. *Bell Syst. Technol. J.* **1975**, *54*, 1451–1462. [CrossRef]

4. Liveris, A.D.; Georghiades, C.N. Exploiting faster-than–Nyquist signaling. *IEEE Trans. Commun.* **2003**, *51*, 1502–1511. [CrossRef]
5. Fan, J.; Guo, S.; Zhou, X.; Ren, Y.; Li, G.Y.; Chen, X. Faster–than–Nyquist signaling: An overview. *IEEE Access* **2017**, *5*, 1925–1940. [CrossRef]
6. Kim, Y.J.D.; Bajcsy, J. Iterative receiver for faster–than–nyquist broadcasting. *Electron. Lett.* **2012**, *48*, 1561–1562. [CrossRef]
7. El Hefnawy, M.; Taoka, H. Overview of faster-than–nyquist for future mobile communication systems. In Proceedings of the 2013 IEEE 77th Vehicular Technology Conference (VTC Spring), Dresden, Germany, 2–5 June 2013; pp. 1–5.
8. Baek, C.U.; Park, G.W.; Jung, J.W. An efficient receiver structure for faster-than-Nyquist signal in MIMO system. *J. Commun.* **2017**, *12*, 285–290. [CrossRef]
9. Rusek, F. A first encounter with faster-than-Nyquist signaling on the MIMO channel. In Proceedings of the 2007 IEEE Wireless Communications and Networking Conference, Kowloon, China, 11–15 March 2007; pp. 1094–1098.
10. Yuhas, M.; Feng, Y.; Bajcsy, J. On the capacity of faster-than-nyquist MIMO transmission with CSI at the receiver. In Proceedings of the IEEE Globecom Workshops (GC Wkshps), San Diego, CA, USA, 6–10 December 2015; pp. 1–6.
11. Bahl, L.; Cocke, J.; Jelinek, F.; Raviv, J. Optimal decoding of linear codes for minimizing symbol error rate. *IEEE Trans. Inf. Theory* **1974**, *20*, 284–287. [CrossRef]
12. Kim, T.H.; Lee, I.K.; Jung, J.W. A study of efficient viterbi equalizer in FTN channel. *J. Korea Inst. Inf. Commun. Eng.* **2014**, *18*, 1323–1329. [CrossRef]
13. Gallager, R.G. Low-density parity-check codes. *Ire Trans. Inf. Theory* **1962**, *8*, 21–28. [CrossRef]
14. Mackay, D.J.C.; Neal, R.M. Near Shannon limit performance of low-density parity-check codes. *Electron. Lett.* **1996**, *32*, 1645–1646. [CrossRef]
15. Lim, B.S.; Kim, M.H.; Jung, J.W. A study on horizontal shuffle scheduling for high speed LDPC decoding in DVB–S2. *J. Korea Inst. Inf. Commun. Eng.* **2012**, *16*, 2143–2149. [CrossRef]
16. Segard, A.; Verdier, F.; Declercq, D.; Urard, P. A DVB-S2 compliant LDPC decoder integrating the Horizontal Shuffle Scheduling. In Proceedings of the 2006 International Symposium on Intelligent Signal Processing and Communications, Tottori, Japan, 12–15 December 2006; pp. 1013–1016.
17. Moose, P. A technique for orthogonal frequency division multiplexing frequency offset correction. *IEEE Trans. Commun.* **1994**, *42*, 2908–2914. [CrossRef]
18. Wu, Y.; Zou, W.Y. Orthogonal frequency division multiplexing: A multi-carrier modulation scheme. *IEEE Trans. Consum. Electron.* **1995**, *41*, 392–399.