Computers and Electrical Engineering 39 (2013) 1980–1992

Contents lists available at SciVerse ScienceDirect

Computers and Electrical Engineering

journal homepage: www.elsevier.com/locate/compeleceng

A new rateless code with unequal error protection property ☆

Ehsan Namjoo [a], Ali Aghagolzadeh [b,*], Javad Museviniya [c]

[a] Faculty of Engineering, Shahid Chamran University of Ahvaz, Ahvaz, Iran
[b] Faculty of Electrical and Computer Engineering, Babol University of Technology, Babol, Iran
[c] Faculty of Electrical and Computer Engineering, University of Tabriz, Tabriz, Iran

ARTICLE INFO

Article history:
Received 17 April 2012
Received in revised form 8 December 2012
Accepted 23 April 2013
Available online 27 May 2013

ABSTRACT

In our previous work, *"robust transmission of scalable video stream using modified LT codes"*, an LT code with unequal packet protection property was proposed. It was seen that applying the proposed code to any importance-sorted input data, could increase the probability of early decoding of the most important parts when enough number of encoded symbols is available at the decoder's side. In this work, the performance of the proposed method is assessed in general case for a wide range of loss rate, even when there are not enough encoded symbols at the decoder's side. Also in this work the degree distribution of input nodes is investigated in more detail. It is illustrated that sorting input nodes in encoding graph, as what we have done in our work, has superior advantage in comparison with unequal input node selection method that is used in traditional rateless code with unequal error protection property.

1. Introduction

In many applications, like network communication, the paths between receivers and transmitters are modeled as erasure channels. The binary erasure channel (BEC), for the first time, was introduced by Elias [1]. Elias showed that the capacity of the BEC with erasure probability p, equals to $1 - p$. He further proved that random codes of rates arbitrarily close to $1 - p$ can be decoded on this channel using maximum likelihood decoding with a small probability of error. There was not a deserving attention to the Elias works in up to 40 years and the BECs were remained just as a theoretical issue. By the vast extension of Internet and computer networks, the Elias's work found its way to practical usages. On the Internet, data are encapsulated into data packets. Each packet has a header that determines the source and the destination of it, and often a sequence of bits describing the absolute or relative position of the packet within the data bitstream. These packets are transmitted on the network from the sender to the receiver(s). Sometimes, packets get lost and never reach to the destination; also packets may be declared lost if their internal checksum does not match. In these cases, the packet is no longer a conforming one. The mentioned conditions make the Internet a very good real-world application for the binary erasure channels.

In an erasure channel, maximum likelihood (ML) decoding of linear codes is equivalent to solving a system of linear equations. This task can be done using Gaussian elimination technique. However, Gaussian elimination is not fast enough, especially when the length of the code is long. Reed-Solomon (RS) codes [2] can be used as an appropriate substitute for erasure channels. RS codes can be decoded from a block with the maximum possible number of erasures. There are also fast encoding and decoding algorithms based on fast polynomial arithmetic [3,4]. Although RS codes have been utilized previously for packet protection of video streaming [5], but because the encoding and decoding complexity significantly increases for long

☆ Reviews processed and recommended for publication to Editor-in-Chief by Associate Editor Dr. Isaac Woungang.

* Corresponding author.
E-mail addresses: e.namjoo@scu.ac.ir (E. Namjoo), aghagol@nit.ac.ir (A. Aghagolzadeh), niya@tabrizu.ac.ir (J. Museviniya).

0045-7906/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.compeleceng.2013.04.015

packets, using them for forward error protection is not an appropriate choice for sources that composed of long length data packets.

In addition to encoding and decoding complexity, there are some other disadvantages for traditional block codes when they come to be used in erasure channels. Estimating the loss rate in an erasure channel just by a fixed erasure rate is not satisfying for the cases in which data is sent concurrently from one sender to many receivers. In that case, the erasure channels from the sender to each of the receivers may have different erasure probabilities. Hence in many applications, the sender may probe the channel to guess a reasonable estimate of the current erasure probability and then it can adjust the coding rate accordingly. But if the number of receivers is large, or in situations such as satellite or wireless transmission where receivers experience sudden unexpected changes in their reception characteristics, it becomes unrealistic to keep track of the loss rates of the individual receivers. So the sender is forced to assume the worst loss rate for all of the receivers. This not only puts unnecessary burdens on the network if the actual loss rate is smaller, but also risks the reliable transmission if the actual loss rate is larger than what is considered.

Tornado codes can be cited as one of the most outstanding efforts to create an efficient code for erasure channels. They could come arbitrarily close to the capacity of the binary erasure channel [6,7]. Like Gallager's LDPC-codes [8], Tornado codes use a highly irregular weight distribution for the underlying graphs. The running times of the encoding and decoding algorithms for tornado codes are proportional to their block-length. That is very outstanding property in comparison with RS codes, but for small rates, the encoding and decoding algorithms for these codes are slow.

LT codes [9] are an important class of forward error correction erasure codes. Whereas traditional erasure codes like RS codes have a fixed code rate that must be chosen before the encoding begins, LT codes are rateless. It means that the encoder can generate as many encoded symbols as needed. This is an advantage that could be used for the channel with unknown or time-varying conditions. The other advantage of an LT code is that the encoder and decoder are not as complicated as those for RS codes. Encoding and decoding complexity for an LT code increases linearly by growing the data length. In addition, any subset of adequate coded symbols could be chosen to recover the whole original source. The later property provides flexibility that makes the code robust against different patterns of loss, i.e., independent of the channel's loss pattern, the decoder just need to receive a specific number of source packets to successfully recover the coded message.

In transmitting data with different importance hierarchies, the protection scenario should be considered in a way that more important parts get higher order of protection. Codes that protect some part of data better than the other parts are called unequal error protection (UEP) codes. In contrast to equal error protection (EEP) codes, where the same level of protection is applied to all information symbols, UEP codes assign different levels of protection to different information symbols. Therefore, in a loss prone channel that the source data are to be protected using LT codes, UEP scenario is considered as the ability of the code to recover the high important data sooner than the other parts. In [10], an UEP LT code is proposed by non-uniformly choosing message bits to encode each symbol. In [11,12], expanding window fountain codes have been applied for multicasting data. The idea of expanding window fountain codes is to encode each symbol based on only source symbols inside a window. The windows are pre-designed in an overlapping and expanding manner such that any larger window contains all the symbols inside a smaller window. However, the designs in [11,12] use a large single code which alter the overall degree distribution and therefore change code behavior significantly [13].

In our earlier work [14], an LT code with unequal packet protection property was proposed and the effectiveness of the proposed scheme in comparison with one of the most brilliant efforts [10], was assessed. In this paper, the main idea to achieve UEP property on LT codes is presented. Then the effectiveness of the proposed method in erasure channels even when there are not sufficient encoding symbols at the decoder's side, is investigated using *reception inefficiency* curves. In [14], two metrics to evaluate our work were utilized. The *performance ratio* metric was used to show the unequal packet recovery property and the *bit error rate versus overhead values* was used to show the unequal error protection property of the new code. In both metrics, each time for a specific value of overhead, the encoding procedure is run and the proportion of the undecoded input source nodes to all source nodes is computed for each set and considered as the bit error rate. But in this work, to show the irrefutable performance of the new code for vast range of erasure, the coding procedure is run just once using a constant value of overhead, and then it is assumed that the number of received output nodes at the decoder side is varying from zero, the worse situation (the erasure probability equals to 1), to the value K + overhead (the erasure probability equals to 0). Each time for any value of received output symbols at the decoder's side, the decoding procedure gets started and at the end of decoding, the number of recovered input symbols is computed for each set. The number of unrecovered input nodes to the number of all nodes in each set is considered as the *reception inefficiency* for that set. Using *reception inefficiency* curves, the performance of the proposed approach is evaluated in erasure channels considering a wide range for erasure probability. The same simulations based on new metric have been done for the method given in [10] and the results have been compared. Moreover, in this paper, the input nodes' degree distribution of the encoding graph is investigated in more detail. It is illustrated that sorting input nodes as what we have done in our work have superior advantage in comparison with the unequal input node selection method that is used in traditional rateless code with unequal error property. it will be shown that the ineffectiveness of the unequal input node selection approaches in recovering the least important part, is due to the fact that in non-uniform selecting procedure, sometimes many input nodes remain uncovered that cause the severe decrease of the code's performance when it comes to protect all parts of data. Such a disability never happens in our proposed scheme.

The remaining of this paper is as follows. To make it self-contained, the most related and the essential materials to our work are given in Sections 2 and 3. Section 2 briefly skims the essential basis of LT codes and Section 3 contains a short review of a well known UEP rateless code proposed in [10] and a brief summary of our previous work [14]. In Section 4, the main goals of the paper are brought up and the results are given. Finally Section 5 concludes the paper.

2. Essentials of LT codes

LT code utilizes sparse bipartite graph to generate codewords. The most distinguishing property of an LT code is that it employs a simple algorithm based on the XOR (exclusive OR) operation to encode and decode messages. The encoding process begins by dividing the message into K blocks of equal length, each block is considered as an input symbol; then a random number, d, $1 \leqslant d \leqslant K$ (K is the number of input symbols), is generated by a random number generator function called robust Soliton distribution (RSD). After that, a subset of d input symbols out of K input symbols is randomly selected. Finally an output symbol is generated by applying XOR operations to the selected input symbols. In the same way, as many output symbols as required are generated. The decoder can retrieve all input symbols if it receives enough, often a bit more than K, encoded symbols. The encoding graph consists of two parts: the input symbol part with K input nodes and the output symbol part that consists of all generated codewords (output nodes). Fig. 1 indicates a very simple encoding graph. Each of the input and output nodes could be bits, or in general case, equal size packets of bits.

The decoding of LT codes is performed according to the belief propagation algorithm. The algorithm is as follows [15]:

Find an output node t_n that is connected to only one input node s_k (If there is not such an output node, the decoding algorithm halts at this point and fails to recover remained input nodes.), and then

i. Set $s_k = t_n$.
ii. Add (operate binary XOR) s_k to all output nodes that are connected to it.
iii. Remove all the edges connected to the source packet s_k.
iv. Repeat (i)–(iii) until all the K input symbols are determined.

Luby has shown that for a decodable LT code, the average degree of the output nodes should be $\ln(K)$ and also the average number of operations needed to decode a message is $K\ln(K)$. To keep these conditions satisfied, he suggested a degree distribution function $\rho(d)$ called Soliton distribution [9]:

$$\rho(d=1) = 1/K, \quad \rho(d) = \frac{1}{d(d-1)} \ \mathit{for}\ \ d = 2, 3, \ldots, K \tag{1}$$

Although Eq. (1), as the degree generator function, satisfies the conditions theoretically, but it increases the probability of decoding failure. To solve the problem, the robust soliton distribution was proposed as Eq. (2) [9]:

$$\mu(d) = (\rho(d) + \tau(d))/Z \tag{2}$$

where $\tau(d)$ and Z are indicated in Eqs. (3) and (4), respectively:

$$\tau(d) = \begin{cases} S/Kd, & \text{for } d = 1, 2, \ldots, (K/S) - 1 \\ \frac{S}{K}\ln(S/\delta) & \text{for } d = K/S \\ 0 & \text{for } d > K/S \end{cases} \tag{3}$$

$$Z = \sum_d \rho(d) + \tau(d) \tag{4}$$

S is the number of degree-one output symbols and a very important parameter called ripple size. And δ is the probability of the decoding failure. Ripple size is computed as bellow:

$$S = c \cdot \ln(K/\delta)\sqrt{K} \tag{5}$$

where c is a constant real value in the range $[0,1]$. The ripple size should not be so large; otherwise, the number of overhead symbols required to reconstruct the whole message is increased. On the other hand, small values for ripple size increase the decoding failure probability; it means that it increases the probability that in some decoding step there might be no other

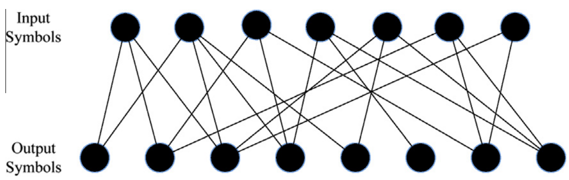


Fig. 1. Simple example of encoding graph.

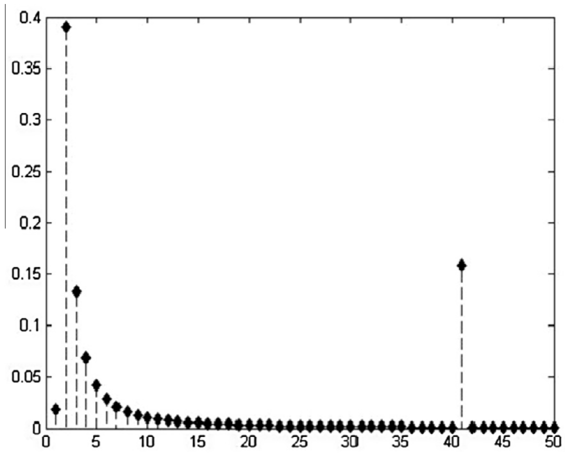


Fig. 2. Robust Soliton probability mass function.

degree-one output symbol to continue the decoding procedure. Robust Soliton function for the values K = 10,000, c = 0.2 and δ = 0.05 is illustrated in Fig. 2.

Luby's analysis explains how designing τ and ρ ensures that the decoding process gets started. Spike shown in Fig. 2 at $d = K/S$ ensure that every source packet is likely to be connected to an output node, at least once.

3. Rateless codes with unequal error protection property

In many studies on rateless codes, equal error protection of all data is considered. The equal error protection property would be appropriate for applications such as multicasting bulk data [16,17]. In some other applications, a portion of data may need to be recovered prior to the rest. An example could be video-on-demand systems, in which some parts of video stream should be reconstructed in sequence [18]. The protection technique, in such scenarios, is often performed using UEP codes.

3.1. Creating rateless codes by non-uniform selecting of input nodes

In [10] authors proposed a rateless code with unequal error protection property. Supposing that there are K input symbols, the input symbols are partitioned into r sets $s_1, s_2, \ldots, s_r$ of size $\alpha_1 K, \alpha_2 K, \ldots, \alpha_r K$ such that $\sum_{i=1}^{r} \alpha_i = 1$. Let $P_i(K)$ is the probability that an edge is connected to a particular input node in s_i for $i = 1, 2, \ldots, r$. it is clear that $\sum_{i=1}^{r} \alpha_i K P_i(K) = 1$. Assuming that $P_i(K) = K_i/K$, the aforementioned equation can be written as $\sum_{i=1}^{r} \alpha_i K_i = 1$.

The encoding procedure for a sequence of n output symbols is as follows:

i. Select randomly a degree d, $d \in \{1, 2, \ldots, K\}$, according to the degree generator function.
ii. Select d distinct input symbols. To select a symbol, at first select one of the r predefined sets, s_i, according to selection probability $P_i(K)$, and then uniformly choose one of the input symbols covered by s_i.
iii. Set an output node equal to bitwise module 2 sum of all selected input nodes.

According to [10], to ensure that the average bit error rate for the input symbols covered by s_i is smaller than the average bit error rate of those from any other set s_j, the probability of selecting an input symbol from s_i should be greater than the probability of selecting input nodes from any other set s_j.

Assuming that there are just two sets s_1 and s_2, the set s_1 is called the set of most important symbols (MIS) and the set s_2 is called the set of least important symbols (LIS). In that case deciding the probability of selecting any symbol from s_1 greater than those from s_2, ensures that the average bit error rate for the symbols covered by set s_1 is smaller than the average bit error rate for elements of the set s_2. To achieved this, $P_1(K)$, the probability of selecting an input symbol from MIS, is set to K_M/K and $P_2(K)$, the probability of selecting an input symbol from LIS, is set to K_L/K where $K_M = \frac{1-(1-\alpha)K_L}{\alpha}$ and $0 < K_L < 1$. The parameter K_M gives the relative importance of symbols from set s_1. Fig. 3 illustrates the encoding procedure for the case that there are two sets.

3.2. Sorted encoding graph LT codes

In this section, our previous effort in designing an UEP erasure code is reviewed. In encoding graph, the input nodes with larger degree are more likely to be decoded earlier [10–12], so modifying the input nodes' arrangement by sorting them

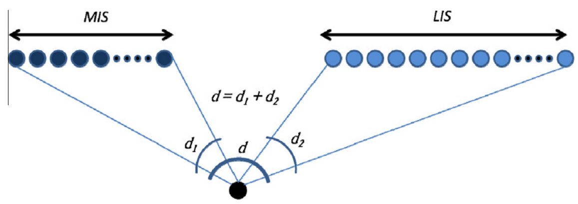


Fig. 3. Non-uniform selecting of input symbols according to RUEP scheme.

according to their degrees could provide UEP property for the input symbols. Fig. 4a shows a simplified example of the encoding graph in its original form (Simple LT code) and Fig. 4b indicates our proposed scheme that the input nodes have been sorted according to their degrees.

By putting any importance-sorted data in the position of the input nodes in the modified graph, the code could protect more important parts better than the other parts. Sorting input nodes according to their degrees is equivalent to the situation that more important data symbols are placed on the place of input nodes with large degree in unsorted original encoding graph. Since the input nodes in the modified encoding graph are sorted by their degrees, by substituting more important symbols as the input nodes at the beginning of the encoding graph, the chance of early decoding of them is increased. Thus, assuming that the importance of a source data increases by moving towards the beginning of the packets, the idea could provide UEP property for any importance-sorted sequence.

4. Simulation results

In noisy channels, a coded symbol is distorted when it passes through the channel and it is polluted by the channel noise in a way that the decoder may decide wrongly about the received symbol, and the wrong symbol may be considered as the right one. In binary noisy channel, when the decoder comes to decode a sequence, error is happened in a way that zeros and ones may be decoded interchangeably. In that case the bit error rate is considered in the range 0–0.5 that the bit error rate equal to 0.5 is the worst case. In many cases like [19,20] authors have used metrics such as bit error rate curves to justify the performance of their suggested coding scheme. But in binary erasure channels the scenario is completely different. In erasure channels any coded symbol is received soundly to the destination or it completely gets lost, so a completely different metric should be used. *Reception inefficiency* as a suitable metric is defined as the proportion of unrecovered symbols to all source symbols at any step of decoding procedure. It is considered in the range of 0–1. *Reception inefficiency* equal to 1 means that none of the source symbols is received, and when it is equal to 0, it means that all the symbols are received soundly. In this case when the *reception inefficiency* is equal to 0.5, it means that the half of source data is received and the rest get lost. Since the most interesting property of a rateless code is its robustness against different patterns of loss, a metric that assesses the performance of a codes according to the received number of encoded symbols at the decoder's side, could make a sound and suitable tool to reflect the performance of the code. In this section, using *reception inefficiency* curves, the effectiveness of the

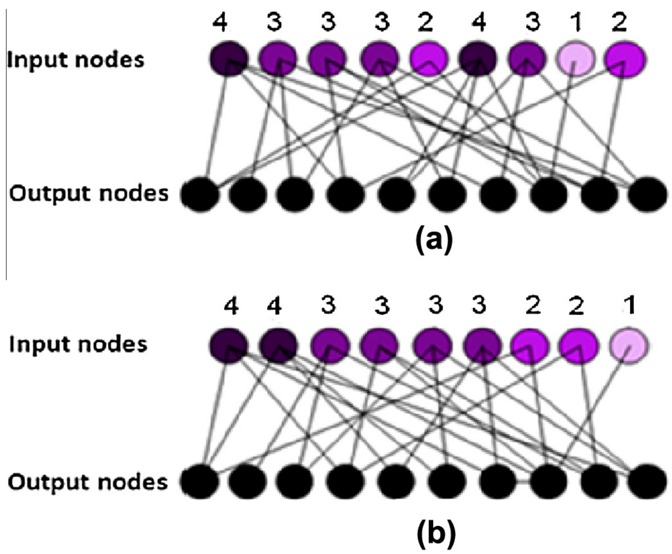


Fig. 4. Encoding graphs: (a) the original form and (b) the proposed modified form.

proposed approach in erasure channels-even when there are not adequate encoding symbol at the decoder's side-is assessed in comparison with a well known previous effort that was proposed by Rahnavard et al. [10]. The basis of the Rahnavard et al. approach was given in Section 3. From now on, this approach is referred as Rahnavard unequal error protection (RUEP) scheme.

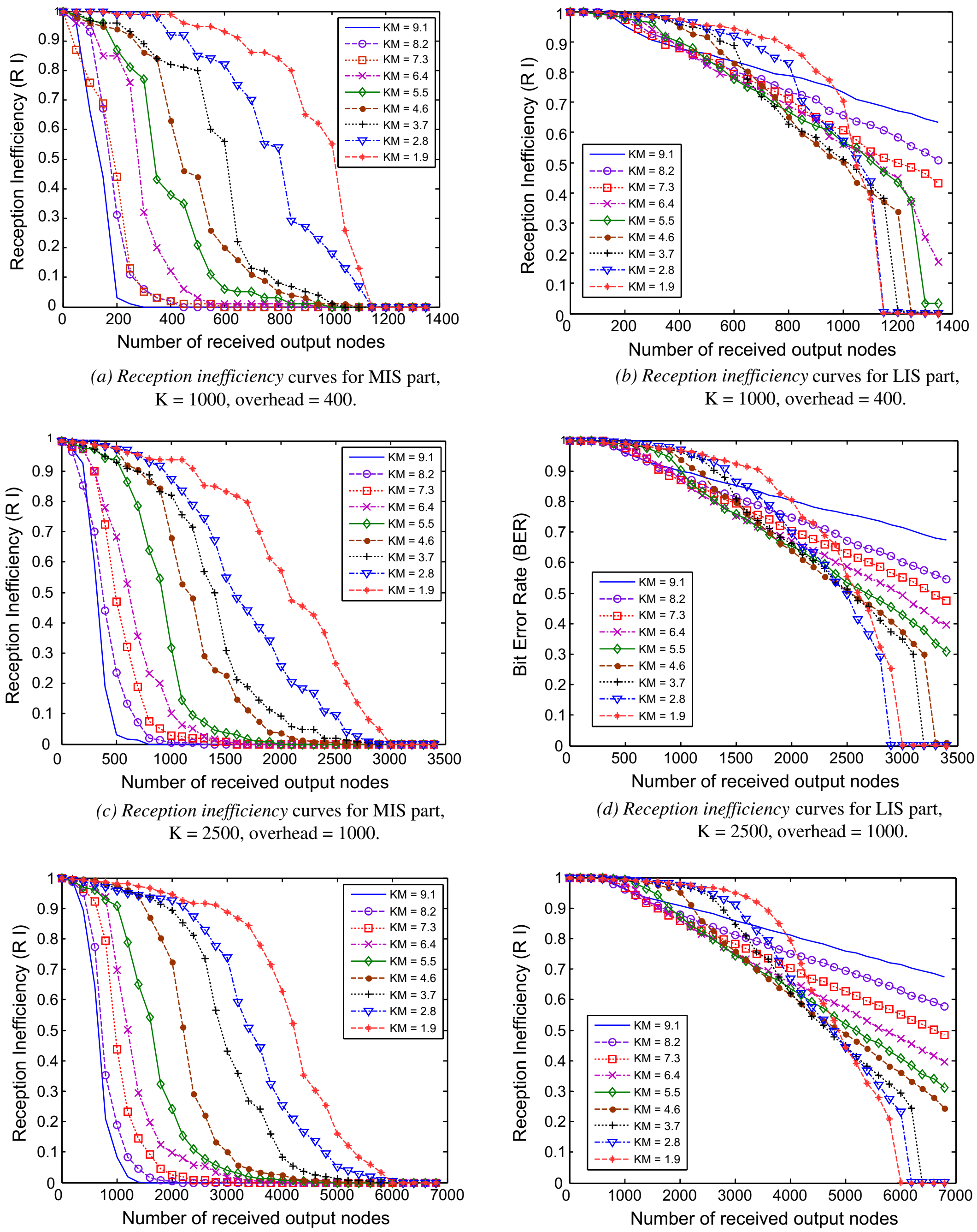
The *reception inefficiency* curves for a given set S, are computed as follows.

(i) Generate $n = K + overhead$ encoded symbols.
(ii) $Count = 0$.
(iii) Select the ($count + 1$) first coded symbol. Run the decoder to decode the selected encoded symbols.
(iv) Compute the reception inefficiency as the number of unrecovered input symbols from set S to the total number of input nodes covered by S.
(v) $Count = Count + 1$.
(vi) Repeat (iii)–(v) until Count is equal to n.

The reception inefficiency curves for RUEP and our proposed approach are illustrated in Figs. 5 and 6. The overhead values are considered equal to 400, 1000, 2000 and 4000 for the cases that the number of input values is 1000, 2500, 5000 and 10,000, respectively. Also α is set to 0.1 for all. So, the number of input symbols in MIS is considered one tenth of the total number of input symbols. The values of c and δ are considered to 0.2 and 0.05 for all the following simulations, respectively.

In Fig. 5a and b, the number of input nodes and the value of the overhead is set to 1000 and 400, respectively. As illustrated in Fig. 5a, for K_M values equal to 9.1, 8.2, 7.3, 6.4 and 5.5, the *reception inefficiency* for MIS set, approaches to zero even when the number of received output node at the decoder side is much less than K. It shows the excellent protection performance of RUEP on MIS for large values of K_M. In other words, the decoder could reconstruct the whole MIS part without using excess overhead nodes. But, as seen in Fig. 5b, for the same values of K_M, the code provides unsatisfactory and even poor results on LIS. For instance, when K_M is equal to 9.1, the whole MIS part could be recovered (the *reception inefficiency* for MIS comes to zero) just by receiving small number of output symbols, but after receiving all generated output nodes (1400 output nodes) the *reception inefficiency* for the LIS hardly approaches to some value around 0.6. It means that, the code could only retrieve 40% of LIS after receiving 40% of the overhead. Also, as shown in Fig. 5c and d, similar results are given when the number of input nodes is increased to 2500. In this case, the *reception inefficiency* performance for LIS shows weaker results in contrast to the previous case. By increasing the number of input nodes to 5000 and 10,000 in Fig. 5e–h, we have similar results. The performance of the code on LIS decreases even as the performance of the code on MIS still is outstandingly good. In order to compare the results with our proposed code, the same simulations have been carried out under the same conditions. Fig. 6 shows the *reception inefficiency* curves for our proposed code. The method of calculating *reception inefficiency* curves is the same as what have been done for RUEP. Comparing the plots illustrated in Fig. 6a with the plots in both Fig. 5a and b, leads the following results. For large values of K_M, RUEP could provide very strong protection on MIS part, and the *reception inefficiency* on MIS is so much better than the results provided by the proposed method. But, as seen in Fig. 5b, for the large values of K_M, the protection quality on LIS is much weaker than what we have for our proposed method. For the other smaller values of K_M, that the protection quality on LIS is comparable with the proposed method, the protection quality on MIS for RUEP does not provide so much better results in comparison with our proposed method. Also, comparing the results given in Fig. 6b with the results given in both Fig. 5c and d leads to similar conclusions. That means when the number of input values is chosen to 2500, the same story happens. Again, for large values of K_M, the performance of RUEP over MIS is much better in comparison with our proposed method, but overall again, for large values of K_M, RUEP provides poor results on LIS. On the other hand, according to the results shown in Fig. 5d, when K_M is chosen equal to 1.9, the protection performance of RUEP on both sets MIS and LIS is better than our proposed method. By selecting the number of input nodes equal to 5000 (Fig. 5e and f), when K_M is set to 1.9, the performance of RUEP on LIS show better results in comparison with the proposed method, but RUEP with other values of K_M do not provide better error protection performance on LIS. As before, most of the time, RUEP provides better protection performance on MIS. Finally by increasing the number of input nodes to 10,000, comparing the results given in Fig. 6d with both Fig. 5g and h provides similar results except that this time the performance of our proposed method on LIS outperforms RUEP approach. For all values of K_M. It means that for all values of K_M, by increasing the number of input nodes to 10,000, the proposed method could recover the LIS part by receiving smaller number of output nodes.

For further investigation, the protection performance of the proposed method is again compared with RUEP scheme in the case that the number of overhead value is considered less than the previous case and is equal to 25% (the previous simulations were performed considering the overhead equal to 40%). The simulation results for RUEP and the proposed method are given in Figs. 7 and 8. When the number of overhead is decreased to 25% of the input nodes, RUEP's protection performance on both sets of MIS and LIS gets decreased; while the performance of the proposed scheme does not noticeably change. Similar to the previous case, when the number of input symbol is set to 1000, the protection performance on MIS is still better than the performance of the proposed method but its protection performance on LIS could not provide such better results except for a case that K_M is set to 1.9. The result obtained by considering the number of input nodes equal to 2500 is the same. RUEP's protection performance on LIS for K_M equal to 1.9 and 2.8 could provide comparable and somehow better results in comparison with our proposed scheme. By considering the number of input nodes to 5000, the proposed scheme's



(a) *Reception inefficiency* curves for MIS part, K = 1000, overhead = 400.

(b) *Reception inefficiency* curves for LIS part, K = 1000, overhead = 400.

(c) *Reception inefficiency* curves for MIS part, K = 2500, overhead = 1000.

(d) *Reception inefficiency* curves for LIS part, K = 2500, overhead = 1000.

(e) *Reception inefficiency* curves for MIS part, K = 5000, overhead = 2000.

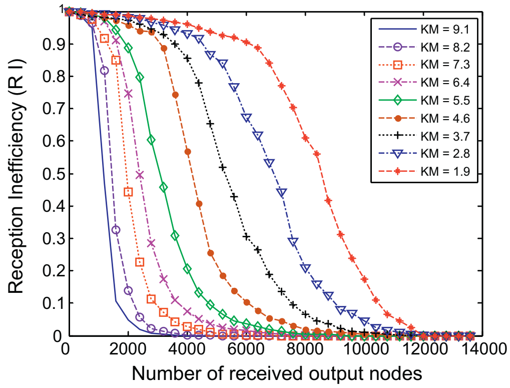
(f) *Reception inefficiency* curves for MIS part, K = 5000, overhead = 2000. ta

Fig. 5. *Reception inefficiency* performance curves for RUEP scheme (40% overhead).

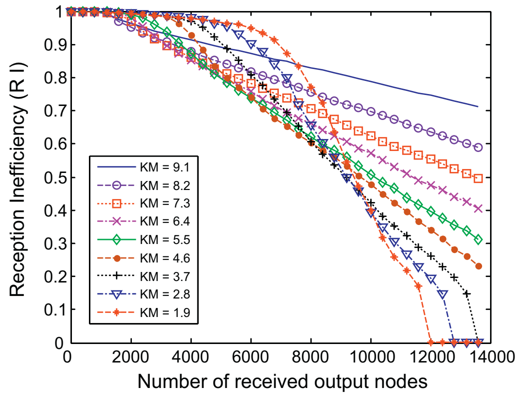
protection performance on LIS outperforms RUEP scheme. Finally, by setting the number of input nodes equal to 10,000, once more RUEP's protection performance on LIS is comparable with the proposed method when K_M is set to 1.9.

According to the results shown in Figs. 5–8, we can say that most of the time, the error protection performance of RUEP scheme on MIS is marvelously good and much better than the proposed scheme especially when the value of K_M is large. But by using large values for K_M in RUEP, the protection performance on LIS is severely decreased.

RUEP encoding procedure provides a condition under which probability of choosing the most important parts of data increases and, as a result, the most important parts participate in producing output nodes much more than LIS symbols do.

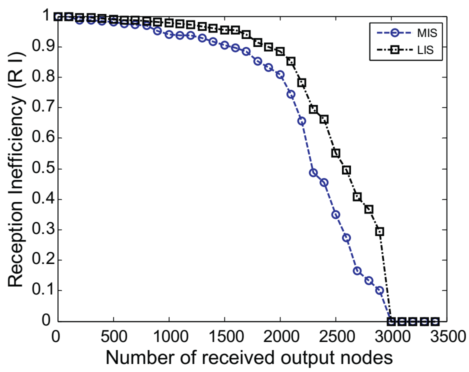


(g) *reception inefficiency* curves for MIS part, K = 10000, overhead = 40000

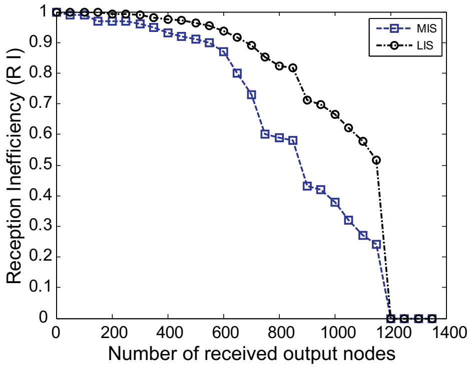


(h) *reception inefficiency* curves for MIS part, K = 10000, overhead = 4000.

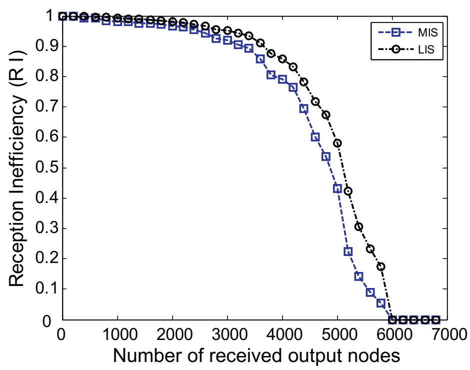
Fig. 5. (*continued*)



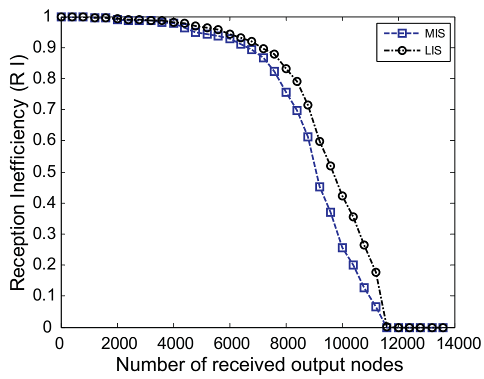
(a) *Reception inefficiency* curves for K = 1000, overhead = 400.



(b) *Reception inefficiency* curves for K = 2500, overhead = 1000.



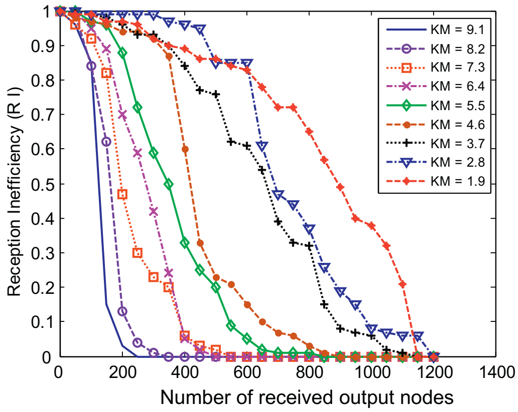
(c) *Reception inefficiency* curves for K = 5000, overhead = 2000.



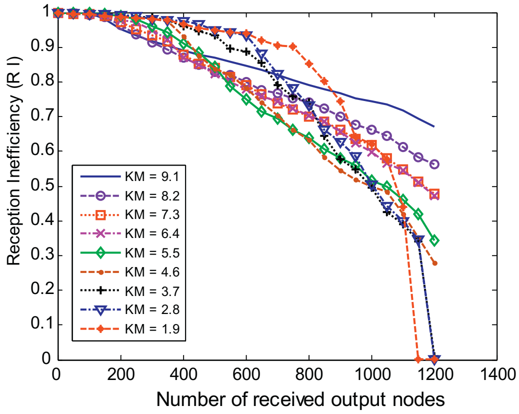
(d) *Reception inefficiency* curves for K = 10000, overhead = 4000.

Fig. 6. *Reception inefficiency* performance curves for the proposed scheme (40% overhead).

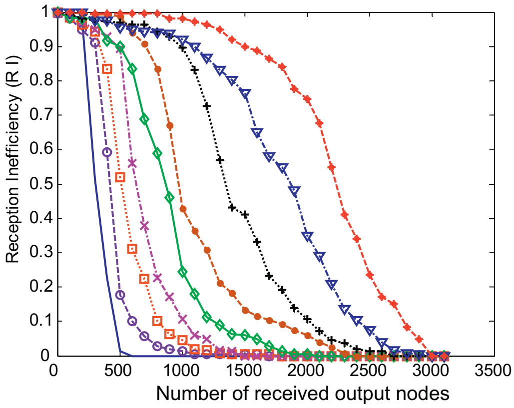
Actually, the symbols from MIS get more protection by decreasing the chance of selecting the symbols from LIS. In other words, the encoder sacrifices the protection performance of symbols from LIS to provide more protection over symbols of MIS; and that's why RUEP performance over MIS is fabulously good. Fig. 9a–i, illustrate the input nodes distributions of the RUEP approach when the input nodes are sorted according to their degrees. In these figures, the number of input nodes is set to 5000 and the overhead is considered equal to 2000. The value of α is set to 0.1 as before. The other parameters are unchanged. As shown in these figures, for large values of K_M, the gap between the degrees of high priority symbols (MIS) and



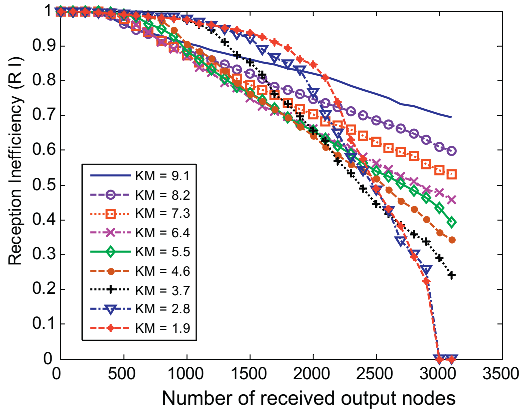
(a) *Reception inefficiency* curves for MIS part, K =1000, overhead = 250.



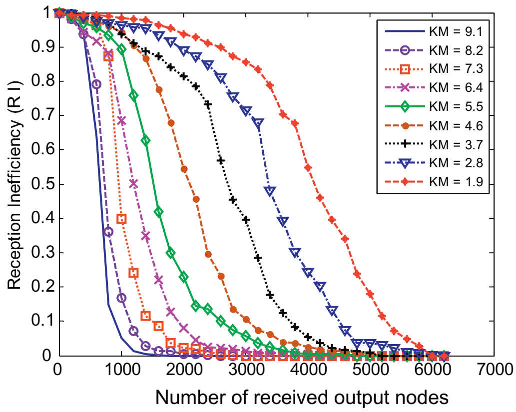
(b) *Reception inefficiency* curves for LIS part, K = 1000, overhead = 250.



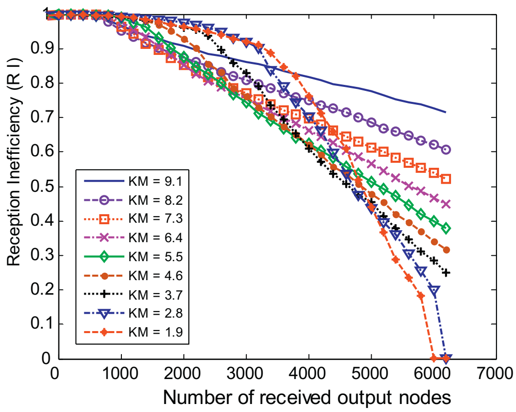
(c) *Reception inefficiency* curves for MIS part, K = 2500, overhead = 625.



(d) *Reception inefficiency* curves for LIS part, K = 2500, overhead = 625.



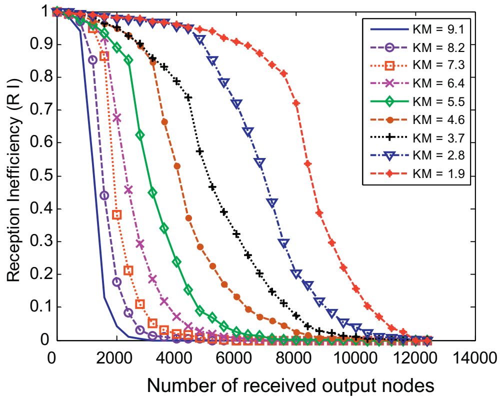
(e) *Reception inefficiency* curves for MIS part, K = 5000, overhead = 1250.



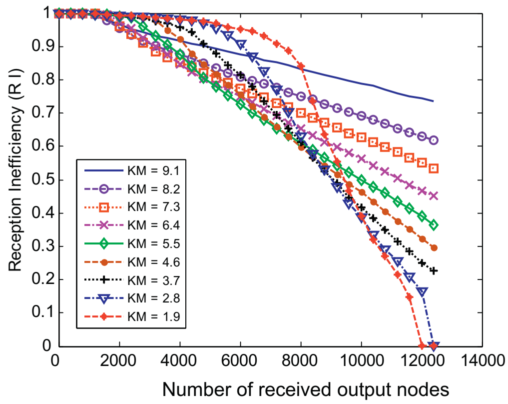
(f) *Reception inefficiency* curves for MIS part, K = 5000, overhead = 1250.

Fig. 7. *Reception inefficiency* performance curves for RUEP scheme (25% overhead).

low priority ones (LIS), is very large. For example, when K_M is equal to 9.1 in Fig. 9a, the minimum degree of symbols in MIS is 80 while the maximum degree of symbols in LIS is 8. This, outstandingly, show that RUEP sacrifices the protection performance of symbols of LIS to provide more protection over MIS symbols. By decreasing the value of K_M, this rapid change among MIL and LIS gets smoothed. The smoothest case is happened when the proposed scheme is used (in Fig. 9j).

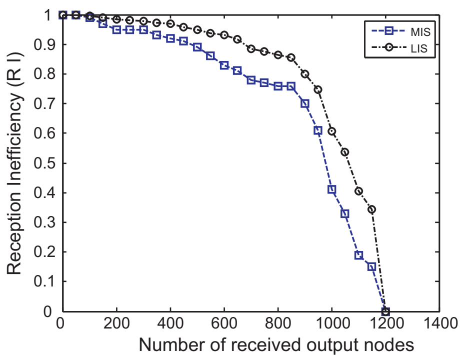


(g) *Reception inefficiency* curves for MIS part, K =10000, overhead = 2500.

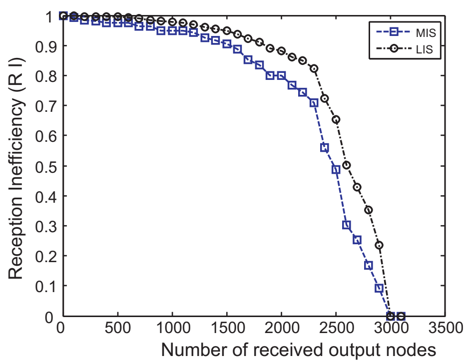


(h) *Reception inefficiency* curves for MIS part, K =10000, overhead = 2500.

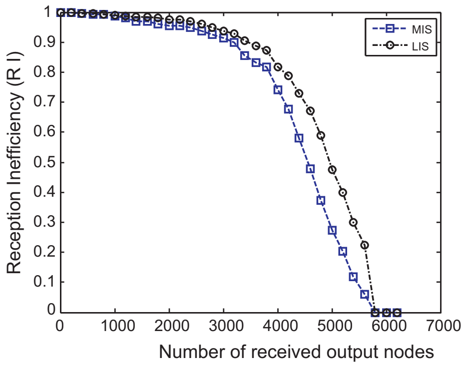
Fig. 7. (*continued*)



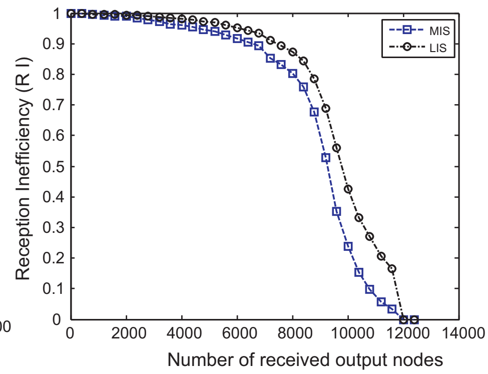
(a) *Reception inefficiency* curves for K = 1000, overhead = 250.



(b) *Reception inefficiency* curves K = 2500, overhead = 625.



(c) *Reception inefficiency* curves for K = 5000, overhead = 1250.

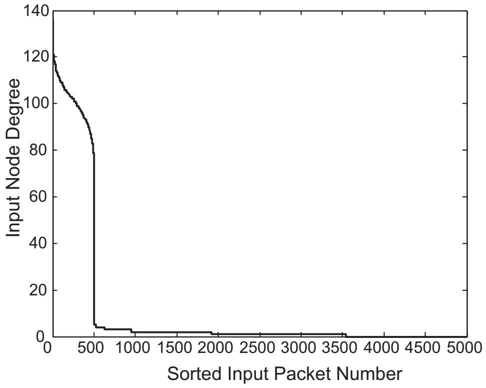


(d) *Reception inefficiency* curves for K = 10000, overhead = 2500.

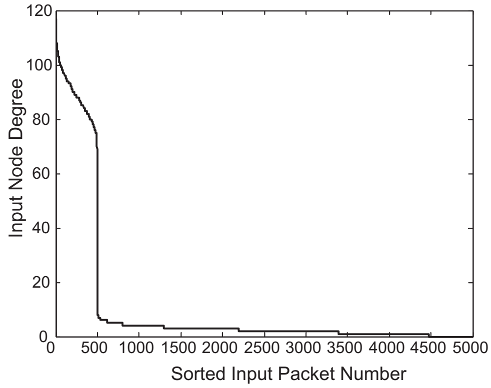
Fig. 8. *Reception inefficiency* performance curves for the proposed scheme (25% overhead).

In our proposed scheme the input nodes' degree distribution is similar to simple LT code. The only difference is that in our proposed scheme, the input nodes are sorted according to their degrees and most important data symbols (MIS part) are placed in the place of input nodes with higher degrees in encoding graph. Thus, in contrast to the RUEP approach, the protection strength that is given to MIS by this simple change in encoding graph, does not weaken the protection performance over LIS part.

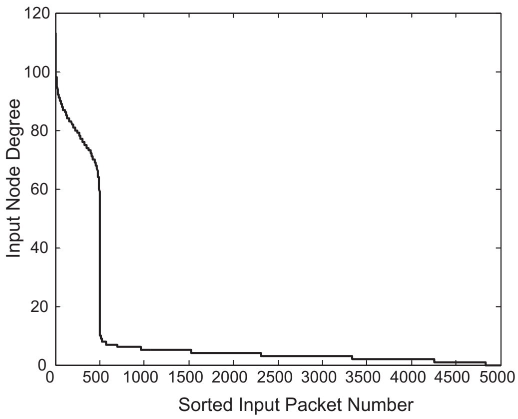
In RUEP, as shown in Fig. 9, for large values of K_M like 9.1, 8.2, 7.5 and 6.4, some nodes on LIS remained uncovered. It means that at the end of encoding procedure their degree remains zero. In that case, it is clear that the decoder is not able



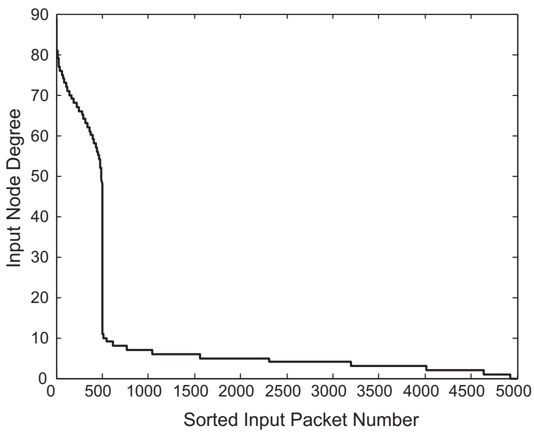
(a) Sorted input degree distribution for RUEP, $K_M = 9.1$.



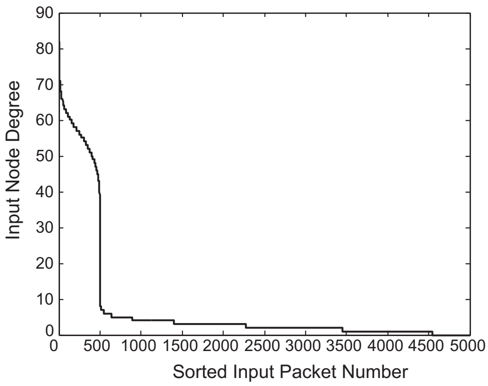
(b) Sorted input degree distribution for RUEP, $K_M = 8.2$.



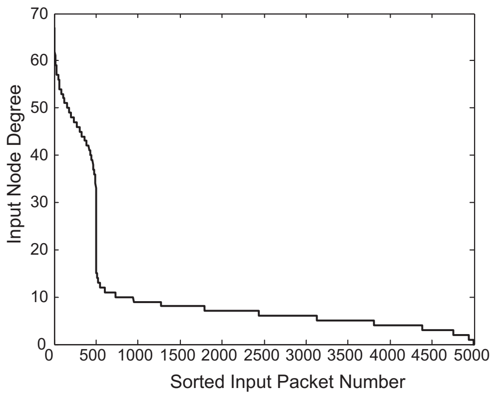
(c) Sorted input degree distribution for RUEP, $K_M = 7.3$.



(d) Sorted input degree distribution for RUEP, $K_M = 6.4$.



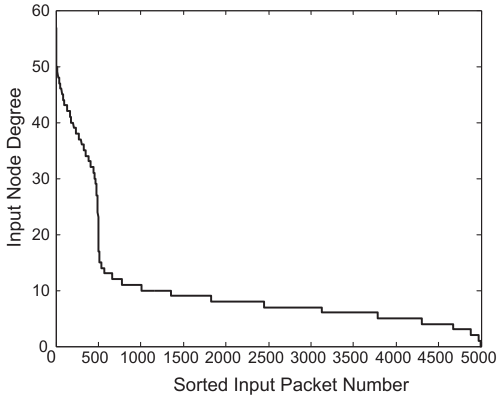
(e) Sorted input degree distribution for RUEP, $K_M = 5.5$.



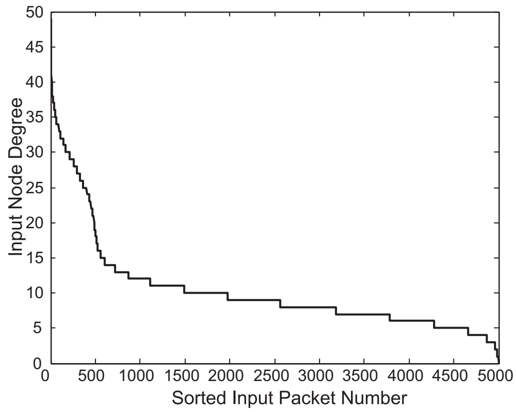
(f) Sorted input degree distribution for RUEP, $K_M = 4.6$.

Fig. 9. Input nodes degree distribution ($K = 5000$, overhead = 2000).

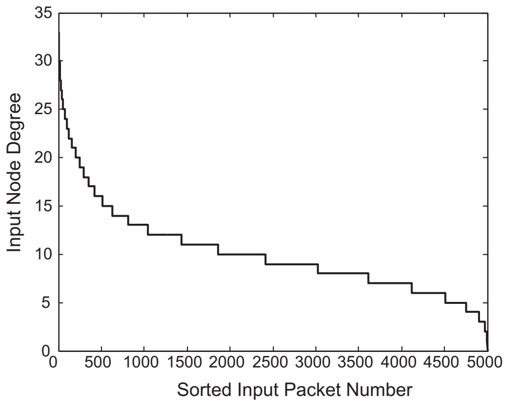
to recover such nodes. Therefore, because of uncovered input nodes, after the specific step, the *reception inefficiency* for LIS may not be decreased to any degree. Hence, using large values of K_M, that brings incredibly good protection on MIS, provides very weak protection performance over LIS part.



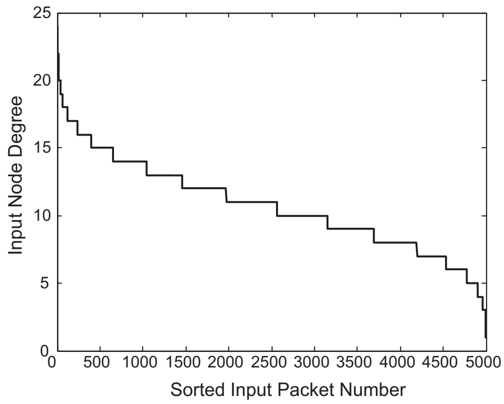
(g) Sorted input degree distribution for RUEP, $K_M = 3.7$.



(h) Sorted input degree distribution for RUEP, $K_M = 2.8$.



(i) Sorted input degree distribution for RUEP, $K_M = 1.9$.



(j) Sorted input degree distribution for simple LT code

Fig. 9. (*continued*)

5. Conclusion

According to the simulation results, our proposed code not only protects the most important parts of data better than the rest, but also it could provide a fairly good protection for the least important part too. Moreover, by studying the output nodes distribution in encoding graphs, we can conclude that the ineffectiveness of the approach given in [10] over LIS, is due to the fact that in a non-uniform selecting procedure, sometimes many input nodes remain uncovered that cause a severe decrease of the code's performance when it comes to protect all parts of data. Such a situation never happens to our suggested approach.

Acknowledgment

This research has been partially supported by Iran Telecommunication Research Centre (ITRC) which is highly appreciated.

References

[1] Elias P. Coding for two noisy channels. In: Proceedings of the 3rd London symposium on information theory, London, UK; 1955. p. 61–76.
[2] Reed IS, Solomon G. Polynomial codes over certain fields. J Soc Ind Appl Math 1960;8:300–4.
[3] Blackburn SR. Fast rational interpolation reed-solomon decoding and the linear complexity profiles of sequences. IEEE Trans Inf Theor 1997;43(2):537–48.
[4] El-Khamy M, McEliece RJ. Iterative algebraic soft-decision list decoding of Reed-Solomon codes. IEEE J Sel Areas Commun 2006;24(3):481–90.
[5] Ahmed T, Mehhaoua A, Boutaba R, Iraqi Y. Adaptive packet protection video streaming over IP networks: a cross layer approach. IEEE J Sel Areas Commun 2005;23(2):385–401.
[6] Luby M, Mitzenmacher M, Shokrollahi A, Spielman D. Efficient erasure correcting codes. IEEE Trans Inf Theor 2001;47(2):569–84.

[7] Luby M, Mitzenmacher M, Shokrollahi A, Spielman D, Stemann V. Practical loss-resilient codes. In: Proceedings of the twenty-ninth annual ACM symposium on theory of computing; 1997.
[8] Gallager RG. Low density parity-check codes. Cambridge (MA): MIT Press; 1963.
[9] Luby M. LT codes. In: The 43rd annual IEEE symposium on foundations of computer science; 2002.
[10] Rahnavard N, Vellambi BN, Fekri F. Rateless codes with unequal error protection property. IEEE Trans Inf Theor 2007;53(4).
[11] Sejdinovic D, Vukobratovic D, Doufexi A, Senk V, Piechocki R. Expanding window fountain codes for unequal error protection. IEEE Trans Commun 2009;57(9).
[12] Vukobratovic D, Stankovic V, Sejdinovic D, Stankovic L, Xiong Z. Scalable video multicast using expanding window fountain codes. IEEE Trans Multimedia 2009;11(6).
[13] Cao Y, Blostein S, Chan W. Unequal error protection rateless coding design for multimedia multicasting. In: IEEE ISIT conference, Austin, Texas; 2010.
[14] Namjoo E, Aghagolzadeh A, Museviniya J. Robust transmission of scalable video stream using modified LT codes. Comput Electr Eng 2011;37(5):768–81.
[15] Mackey JC. Fountain codes. IEE Proc Commun 2005;152:1062–8.
[16] Bayer JW, Luby M, Mitzemmacher M. A digital fountain approach to asynchronous reliable multicast. IEEE J Sel Areas Commun 2002;20.
[17] Mitzenmacher M. Digital fountains: a survey and look forward. In: Proceedings of information theory workshop; October 2004.
[18] Xu L. Resource-efficient of on-demand streaming data using UEP codes. IEEE Trans Commun 2003;51(1):63–71.
[19] Halunga SV, Vizireanu DN, Fratu O. Imperfect cross-correlation and amplitude balance effects on conventional multiuser decoder with turbo encoding. Digit Signal Process 2010;20(1):191–200.
[20] Halunga SV, Vizireanu DN. Performance evaluation for conventional and MMSE multiuser detection algorithms in imperfect reception conditions. Digit Signal Process 2010;20(1):166–78.

Ehsan Namjoo received PhD degree in communication systems from University of Tabriz, Tabriz, Iran, in 2011. He joined Shahid Chamran University of Ahvaz, Ahvaz, Iran, as an assistant professor in 2012. His research interests include signal processing, theory of coding and pattern recognition.

Ali Aghagolzadeh received the Ph.D. degree from Purdue University, West Lafayette, IN, USA, in 1991 in electrical engineering. He is currently a professor of Faculty of Electrical and Computer Engineering in Babol University of Technology, Babol, Iran. His research interests include image processing, video coding and compression, information theory, and computer vision.

Javad Musevi Niya received his Ph.D. degree in communications from University of Tabriz, Tabriz, Iran. Since September 2006, he has been with the Faculty of Electrical and Computer Engineering of the University of Tabriz. His current research interests include wireless communication systems, multimedia networks and signal processing for communication systems.