Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 5546154, 8 pages
https://doi.org/10.1155/2021/5546154

Research Article

A Novel LT Scheme without Feedback Messages for IoT of Smart City Scenarios

Shuang Wu,[1] **Qingyang Guan**,[1,2] **and Shanshan Li**[3,4]

[1] *College of Engineering, Xi'an International University, Xi'an, China 710070*
[2] *College of Electronic and Information Engineering, Shenyang Aerospace University, Shenyang, China 110136*
[3] *College of Electrical and Information Engineering, Huaihua University, Huaihua, China 418000*
[4] *Key Laboratory of Intelligent Control Technology for Wuling-Mountain Ecological Agriculture in Hunan Province, Huaihua 418000, China*

Correspondence should be addressed to Shuang Wu; wooshuang@126.com

Received 25 February 2021; Accepted 23 April 2021; Published 8 May 2021

Academic Editor: Xin Liu

For IoTs of smart city scenarios always with the low cost, low power consumption, and high transmission delay properties, the traditional protocols based on feedback messages, e.g., the Automatic Repeat reQuest (ARQ) schemes, would dramatically affect the transmission efficiency. Therefore, the LT codes with only one feedback message in each entire coding process can be used to substitute the traditional protocols. As in many IoTs of smart city scenarios, the data must have both high transmission efficiency and timeliness requirements; thus, the negative effect of only the feedback message in each entire coding process cannot be neglected in such transmission environments. To enhance the transmission efficiency of such ensembles, a novel LT scheme without feedback messages is proposed in this paper. By presenting the definitions of optimal decoding overhead and recovery ratio per symbol, the optimal decoding overhead of LT codes can be found directly, then the encoding overhead of the encoder can be predesigned also. For this reason, the feedback messages in LT schemes can be removed. By using the proposed LT scheme, the transmission efficiency of IoT of smart city scenarios can be enhanced.

1. Introduction

Being one of the scenarios of massive Internet of Things (IoT), smart cities have received a lot of attention. The IoTs of a smart city always have features such as low cost, low power consumption, small data, and a large amount of nodes [1]. These features make access technology foundations one of the most important foundations of the IoT of a smart city [2].

The Low Power Wide Area Network (LPWAN) is one of the most popular access technologies of the IoTs of a smart city, which can extend the access area into tens of kilometers, and in which the wide area is provided at the price of low power, small data, high delay, etc. Some well-known IoT networks such as Lora, Narrow Band IoT (NBIoT), and SIGFOX are all included in LPWAN [3].

Recently, most of the researches on LPWAN are focused on access technologies, especially on the Nonorthogonal Multiple Access (NOMA), as the NOMA technology can provide a much higher amount of users access into IoTs [4]. The large number of users leads to many problems. The most typical problems are more complex transmission environments and the small size of data for each user [5, 6].

The feature of LPWAN makes the traditional error control codes hard to use to guarantee the reliability of the data in the IoT of a smart city. In fact, in most of the LPWAN networks, the reliability of the data is protected by using the retransmission mechanisms. As the high delay feature exists, the transmission efficiency of these data is hard to guarantee [7, 8]. For this reason, we focused on the rateless codes, which have the potential to provide higher transmission efficiency than the traditional retransmission mechanisms in the IoTs of smart city scenarios.

Rateless codes are a class of error-control codes with an unfixed code rate. In other words, the code rate can be practically as large as needed. LT codes [9], raptor codes [10], and spinal codes [11] are well-known examples of such codes. LT codes are the first practical rateless codes which could provide a capacity approach property on the binary erasure channel (BEC) with encoding/decoding complexity $O(k \log k)$. Raptor codes are the extension of LT codes in which the input symbols of LT codes have been decoded by a high-rate low-density parity check (LDPC) code before the LT encode process begins. The spinal codes are a special class of LT codes which are encoded based on *random hash function*. As LT codes are the basement of both the raptor and spinal codes, our studies pay more attention to it.

For a LT code, the decoder can send a feedback message to the encoder as there are enough output symbols (encoded packets) that have been collected, and the encoding process would be terminated as the message is received. Compared with the traditional protocols which are based on ARQ schemes, the LT codes have much fewer feedback messages. As feedback messages would lead to transmit delay, the LT code can provide greatly higher transmission efficiency than the compared protocols. Unfortunately, as in the smart city scenarios, transmission delay is large; even if the feedback channels do not exist, the only feedback message of each entire LT coding process would also impact the transmission efficiencies [12].

Aiming to provide higher transmission efficiency, and overcome the drawbacks of higher transmission delay and limited energy consumption, we propose a novel LT scheme without feedback messages. As the channel states (packet loss probability) can be estimated [13, 14], feedback messages in the proposed scheme also can be eliminated. By finding out the overhead of a LT code that corresponds with the optimal coding efficiency, the proposed scheme can provide almost optimal transmission efficiency in IoTs of smart city scenarios.

The paper is organized as follows. In Section 2, we review the LT codes, and the asymptotic analysis tool termed And-Or Tree analysis is also given [15]. We analyzed the redundancy probabilities of output symbols in the BP decoding process in Section 3. Then, the proposed LT scheme with optimal decoding overhead is introduced in Section 4, and a series of definitions and lemmas are also provided. Section 5 shows numerical results of the proposed LT scheme; the comparisons of the symbol error rate, recovery rate, and recovery ratio per symbol are all given. And the conclusion of this paper is drawn in Section 5.

2. Preliminaries

In this section, we will briefly review the LT codes, including the encoding and decoding processes, and the well-known And-Or Tree analysis will also be reviewed.

2.1. LT Codes. First, a brief review of LT codes will be provided in the following. LT codes were proposed by Luby, which have been named as the Luby Transform (LT) codes [9]. At the beginning, LT codes were designed to solve the packet loss problems in the erasure channels. For this reason, LT codes cannot be arbitrarily regarded as a class of Forward Error Control (FEC) codes but as a class of erasure correct codes.

LT codes are a class of erasure correct codes, which means the encoding and decoding processes of LT codes are faced to not only the bits but also the packets. For simplicity, in this paper, LT codes are defined to face the symbols; each symbol could be a bit or a packet. The original data is structured in a series of input symbols, and the encoded data is a series of output symbols. In the encoding process of LT codes, the degree distribution of output symbols are necessary, which is defined as $\Omega(x) = \sum_d \Omega_d x^d$, where d is the degree of each output symbol, and Ω_d is the probability of an output symbol with degree d. The encoder of LT codes works as follows: Firstly, the encoder chooses a degree d randomly from the given degree distribution, then selects d input symbols, by summing these input symbols by XOR; an output symbol with degree d is generated. It is worth noting that an output symbol and a corresponding input symbol in a bipartite graph are neighbors of each other. For a LT code with k input symbols, and n denoting the number of output symbols, the overhead is defined as $\gamma = n/k$.

The most common decoding algorithm of LT codes is the Belief Propagation (BP) algorithm. The BP decoding process can be illustrated as follows: At the beginning, the decoder selects an output symbol with degree 1, then the neighbor of this output symbol can decoded. By summing the decoded input symbol on its other neighbors, the degrees of all these neighbors are minus 1, and the selected output symbol and its neighbor are moved out of the decoding process. The BP decoding process can be repeated until there is no output symbol with degree 1 that exists or there are enough input symbols decoded; the decoding processes are called as fail if there is no output symbol with degree 1 that can be found. On the other hand, the decoding processes are called as success.

2.2. And-Or Tree Analysis for LT Codes. The And-Or Tree analysis is a well-known technique that can be used to evaluate the performance of rateless codes [15]. The principle of And-Or Tree analysis is to use an iterative method to approach the symbol error rate of a given LT code.

Consider a LT code with output degree distribution $\Omega(x) = \sum_{d=1}^{D} \Omega_d x^d$, in which the D denotes the maximum degree in the output degree distribution, and it is easy to find that $D \le k$. Similarly, the input degree distribution of this code is given by $\Lambda(x) = \sum_{d=1}^{D'} \Lambda_d x^d$; D' represents the maximum degree of the input symbols, where $D' \le n$. As the BP decoding process can be considered as sending messages "0" or "1" through the edges in a decoding graph between adjacent input and output symbols, only if an edge can be eliminated, then "1" is sent on it. Aimed at constructing the iterative expression, the BP decoding process is divided into several rounds. In each round, input symbols send messages to output symbols, and then, output symbols send messages to input symbols. For an input symbol, if it has not been recovered, then message "0" would be sent; if it has already been recovered, then send "1." For an output symbol, if the adjacent input symbols could be recovered by using this output symbol, then a message "1" would be sent; for the other cases, "0" would be sent. Let y_l be the probability that an input symbol sends 0 at round l; in other words, y_l is the error probability of input symbols at round l. Then, we could construct iterative expressions as

$$y_l = f(y_{l-1}), \tag{1}$$

For the purpose of starting the iterative process, the expression $f(x)$ must be given. For each round l, an input symbol could be regarded as the root of a tree with two layers; the children of this root indicated the output symbols connected with it, and the leaf nodes are the input symbols connected with the children. Then, for an arbitrarily chosen child, the probability this child can recover the root is $\omega(1-y_{l-1})$, and for the root, y_l can be expressed by $\lambda(1-\omega(1-y_{l-1}))$; we have

$$y_l = \lambda(1-\omega(1-y_{l-1})), \tag{2}$$

where $\lambda(x)$ and $\omega(x)$ are the probability distribution of a randomly chosen edge connected with an input symbol and an output symbol with each degree, respectively.

As ω_i means the probability that a randomly chosen edge connected with an output symbol with degree $i+1$, then we have $\omega_i = (i+1)\Omega_{i+1}/\Omega'(1)$. Similarly, the probability that a randomly chosen edge connected with an input symbol with degree $i+1$ can be given as $\lambda_i = (i+1)\Lambda_{i+1}/\Lambda'(1)$.

3. The Redundancy Analysis of LT Codes

For each LT code, as the amount of the input symbols is always less than that of the collected output symbols, there are some output symbols that can be considered as *redundant*, and the *redundancy probability* can be used to measure the margin between the amounts of recovered input symbols and the collected output symbols.

In the BP decoding process of a LT code, an output symbol with degree d is redundant and can be divided into two cases: One case is where all the related input symbols are recovered by using other output symbols, and it is obvious to see that the output symbol is not necessary. The other case is where there are less than $n-1$-related input symbols that can be decoded by using other output symbols, as one output symbol cannot recover more than one input symbol in the meantime, in which the output symbol would also lead to a higher redundancy probability. To distinguish these two cases of redundancy, the output symbols in the first case are deemed as *repeated*, and the second case is named as *unreleased*. By contrast, if there is an output symbol, whose related input symbols have all been recovered, the output symbol is termed as *released*. It is easy to see that the repeated is a particular case of released.

3.1. Redundancy Probabilities of Output Symbols. In a given LT code, assuming there are k input symbols and n output symbols, then the overhead is $\gamma = n/k$.

For an LT code, by given an overhead γ, there is a corresponding symbol error rate y that exists, then the redundancy probability of an output symbol can be given by the following lemma.

Lemma 1. *For an output symbol with degree d, whose redundancy probability can be computed by*

$$P_{d,rdu} = 1 - d(1-y)^{d-1}y. \tag{3}$$

Proof. It is obvious, for the output symbol, the unreleased probability can be given by

$$P_{d,\text{unrel}} = 1 - d(1-y)^{d-1}y - (1-y)^d. \tag{4}$$

And the probability this symbol is repeated is

$$P_{d,\text{rep}} = (1-y)^d. \tag{5}$$

Then, the redundancy probability of this symbol is

$$P_{d,\text{rdu}} = P_{d,\text{unrel}} + P_{d,\text{rep}} = 1 - d(1-y)^{d-1}y. \tag{6}$$

Hence, Lemma 1 is proven.

3.2. Release Probability of Output Symbols. Consider an output symbol with initial degree d, whose degree would be gradually reduced; if this symbol can be released, its degree would be reduced to "1" or "0"; it is worth noting that the degree reduced to "1" means there is an input symbol that can be recovered by this output symbol, and "0" represents all the related input symbols that are recovered by using the previously output symbols.

Let $P_{d\to\widehat{d}}$ denote the probability the degree of this output symbol is reduced to $\widehat{d}(\widehat{d} \le d)$. As a symbol error rate y is given, $P_{d\to\widehat{d}}$ can be computed by

$$P_{d\to\widehat{d}} = \begin{pmatrix} d \\ \widehat{d} \end{pmatrix} (1-y)^{d-d\wedge} y^{d\wedge}. \tag{7}$$

Lemma 2. *The released probability of an output symbol with degree d can be given by*

$$P_{d,rel} = dy(1-y)^{d-1} + (1-y)^d. \tag{8}$$

Proof. It can be found that if the degree of the output symbol can be reduced to "1" or "0," this symbol is released. Thus, the released probability of this symbol can be computed as

$$\begin{aligned} P_{d,\text{rel}} &= P_{d\to 1} + P_{d\to 0} \\ &= \begin{pmatrix} d \\ 1 \end{pmatrix} (1-y)^{d-1} y^1 + \begin{pmatrix} d \\ 0 \end{pmatrix} (1-y)^{d-0} y^0 \\ &= dy(1-y)^{d-1} + (1-y)^d. \end{aligned} \tag{9}$$

Then, equation (8) is obtained.

4. The LT Codes with Optimal Decoding Overhead

For the traditional error control codes, the transmission efficiency can be illustrated by using the code rate, and the higher code rate would lead to a higher transmission efficiency. For LT codes, which are the first class of rateless codes, the definition of the code rate does not exist, instead of overhead. The encoder of a LT code can generate output symbols as much as is needed, then the code rate which is defined as a fixed value cannot be given. Different with the code rate, the overhead is defined as a dynamic ratio but not a fixed value. For this reason, the *decoding overhead* is proposed to denote the value of overhead when the decoding process is finished. As the decoding overhead is a fixed value, based on the definition of overhead, the decoding overhead is the reciprocal of the code rate. And the decoding overhead can be used to measure the transmission efficiency of given LT codes.

4.1. The Quantization of the Transmission Efficiency of LT Codes. To quantize the transmission efficiency of LT codes, a definition named as *Recovery Ratio Per Symbol* (RRPS) is proposed. For a given LT code with k input symbols, the output degree distribution is $\Omega(x)$. For a given decoding overhead γ, the related decoding overhead is $y(\gamma)$. Let $\beta\gamma$ represent the RRPS, then the RRPS is defined as

$$\beta(\gamma) \triangleq \frac{\text{the number of recovered input symbols}}{\text{the number of output symbols}} = \frac{1-y(\gamma)}{\gamma}. \tag{10}$$

To use the Monte Carlo method, it is easy to find out that there is a maximum value of RRPS, but which is helpless to quantize the transmission efficiency of a LT code. Following with the analysis method in Section 3, equation (10) can be given in the following lemma.

Lemma 3. *The RRPS of a given LT code can be represented by the ratio between the number of unredundant output symbols and the total number of output symbols and which can be given by*

$$\beta(\gamma) = y(\gamma)\omega(1-y(\gamma)). \tag{11}$$

Proof. RRPS is the ratio between the number of unredundant output symbols and the total number of output symbols, which is defined as

$$\beta(\gamma) \triangleq \frac{\text{the number of unredundant output symbols}}{\text{the number of output symbols}} = \frac{\sum_{d=1}^{D}\Omega_d \gamma k(1-P_{d,\text{red}})}{\gamma k} = \sum_{d=1}^{D}\Omega_d d(1-y(\gamma))^{d-1}y(\gamma). \tag{12}$$

As $\Omega(x)=\sum_{d=1}^{D}\Omega_d x^d$ and $\Omega'(x)=\omega(x)=\sum_{d=1}^{D} d\Omega_d x^{d-1}$, then equation (11) is obtained.

As Lemma 3 is given, then the following lemma can also be given.

Lemma 4. *For a given LT code, the RRPS is the only maximum value in the domain of definition of γ.*

Proof. By observing equation (12), it can be found that $y(\gamma)$ is a monotone decreasing function of variate γ, and the value range is $y(\gamma)\in(0,1)$. For the same reason, $\sum_{d=1}^{D}\Omega_d d(1-y(\gamma))^{d-1}$ is a monotone increasing function of variate γ, and the value range is $(0,\sum_{d=1}^{D} d\Omega_d)$ with the same domain of definition of $y(\gamma)$.

$y(\gamma)$ and $\sum_{d=1}^{D}\Omega_d d(1-y(\gamma))^{d-1}$ are the monotone decreasing and increasing functions in the same domain of definition, and they are not the reciprocal function for each other; thus, their product is the only one extreme value.

Let $\phi(\gamma)=1-y(\gamma)$, then equation (10) can be rewritten as $\beta(\gamma)=\phi(\gamma)/\gamma$. Then, the derivative and section derivative functions of the RRPS are given as

$$\begin{aligned}\beta'(\gamma) &= \frac{\gamma\phi'(\gamma)-\phi(\gamma)}{\gamma^2},\\ \beta''(\gamma) &= \frac{\gamma^3\phi''(\gamma)-(\gamma^2+1)\phi'(\gamma)-2\gamma\phi(\gamma)}{\gamma^4}.\end{aligned} \tag{13}$$

As there is an extreme value of RRPS that exists, let Γ be the value of the overhead which corresponds to the extreme value of RRPS; it is can be given that $\beta'(\Gamma)=0$. The recovered input symbols are always less than the output symbols, which means $\phi(\gamma)$ is a concave function in the domain of γ. Furthermore, when $\phi(\gamma)>0$ and $\phi'(\gamma)>0$ in the given domain, the second derivative of the recovery rate of input symbols satisfies $\phi''(\gamma)<0$. Then, the second derivative of RRPS satisfies $\beta''(\Gamma)<0$. For this reason, $\beta(\Gamma)$ is the maximum value of RRPS.

Thus, Lemma 4 is proven.

As the RRPS directly represents the transmission efficiency of a LT code, the Γ is the value of overhead which corresponds to the optimal transmission efficiency. Thus, the decoding overhead with value Γ is termed as the *optimal decoding overhead.*

4.2. The Symbol Error Rate of LT Codes without Feedback Messages. In earlier studies of LT codes, an encoding process of LT codes can be terminated only if the feedback message is received. As a feedback message would be transmitted to return to the encoder until the symbol error rates are low enough, the encoding process cannot be finished at the moment that the symbol error rate just reaches the low enough level. Actually, the encoder would continue generating output symbols until the feedback message is received. In other words, the overheads of traditional LT codes would be larger than those which are needed to make sure the symbol error rates are low enough; for the scenarios with high transmission delays, the gaps between the practice overheads and

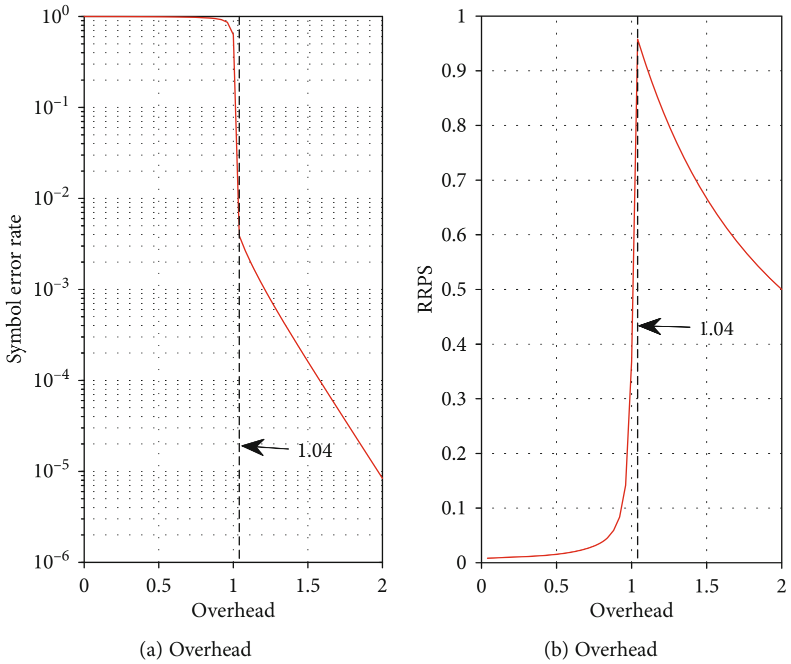

Figure 1: Symbol error rate and RRPS performances of the given LT code.

needed overheads would be much larger. Hence, transmission efficiency of traditional LT codes cannot be guaranteed in IoTs of smart city scenarios.

For traditional LT codes, the symbol error rate can be determined by overhead, which means the symbol error rates would be continually decreasing with overhead growth. Although the symbol error rates of LT codes can reach an arbitrary low level if the overheads are large enough, the transmission efficiency of such a scheme would be dramatically affected. Actually, by finding out the optimal decoding overhead of a LT code, for the different reliability requirements, the symbol error rate could be guaranteed by adjusting the other parameters except the overhead. For this reason, in the proposed LT scheme without feedback messages, the various symbol error rates would be provided by using different output degree distributions.

As the different symbol error rates of different portions of LT codes can be provided by using different selection probabilities in the encoding processes [16, 17], then we can provide the following lemma.

Lemma 5. *Assuming the output degree distribution of a LT code is optimal, then the symbol error rate performance of this code would be increased as the average output symbol degree grows.*

Proof. Consider LT codes with a decoding symbol error rate y, then for an input symbol with degree d, the error rate of this symbol can be computed as

$$y_d = (1 - \omega(1 - y))^{d-1}. \tag{14}$$

It is easy to see that the error rate of this symbol would decrease as degree d grows.

Generalized to all the input symbols, it is not hard to find that the symbol error rate of an input symbol would be lower as it has been selected more times in the encoding process. In other words, symbol error rates of LT codes would increase with the average output symbol degree growing.

5. Experimental Results

In this section, the symbol error rate and recovery ratio per symbol (RRPS) performances of LT codes are presented. By illustrating the symbol error rate and RRPS of the same codes, the transmission efficiencies and optimal decoding overheads can be observed. Then, we compared the proposed optimal decoding overhead with other notations which are designed to measure the corresponding decoding overhead, and the comparison of transmission efficiencies among the proposed and other related schemes in IoT smarty city scenarios are given also.

Firstly, the symbol error rate and RRPS performances of a given LT code are provided in Figure 1 in which the LT code adopted the output degree distribution which was proposed by [10], and which is given in the following:

$$\begin{aligned}\Omega(x) = {} & 0.007969x^1 + 0.493570x^2 + 0.166220x^3 + 0.072646x^4 \\ & + 0.082558x^5 + 0.056058x^8 + 0.037229x^9 \\ & + 0.055590x^{19} + 0.025023x^{64} + 0.003137x^{66}.\end{aligned} \tag{15}$$

Figure 1 illustrated the symbol error rate and RRPS performances of the given code. It is easy to see that the symbol error rate and RRPS performances of the given code are shown in Figures 1(a) and 1(b), respectively. And it is obvious to find out from Figure 1(b) the maximum value of RRPS related to the overhead $\gamma = 1.04$, which means that $\max\{\beta(\gamma)\} = \beta(\gamma)|_{\gamma=1.04}$, then the optimal decoding overhead of the given code is $\Gamma = 1.04$. By contrast, in Figure 1(a), the overhead $\gamma = 1.04$ belongs to the region in which the symbol error rate slowly decreases with the overhead growth, in which the region is well-known as a waterfall region, and it is hard to identify the value of the optimal decoding overhead.

Beside the classical output degree distribution, we also compared the symbol error rate and RRPS performances of LT codes with the robust degree distributions [9] in Figure 2, in which the parameters of robust degree distributions for the codes 1 and 2 are $(k = 1000, c = 5, \delta = 0.1)$ and $(k = 1000, c = 0.01, \delta = 0.01)$, respectively. By computing using output degree distributions, these two codes are with the average output degree $\bar{d}_1 = 8.282$ and $\bar{d}_1 = 10.761$, respectively. One can easily find that code 2 has a better symbol error rate performance than code 1, which means the codes with higher average output degrees can provide better symbol error rate performances than the ones with lower average output degrees, but the higher average output degrees would also lead to higher encoding and decoding complexities than the other ones. The RRPS performances of these two codes are given in Figure 3; it can be found that although code 2 has a better symbol error rate performance than code 1, the transmission efficiency of these two codes are the same, and the optimal decoding overheads of these two codes are both at $\Gamma = 1.08$.

We also use the RRPS to illustrate the transmission efficiency of some other LT-based codes and compared with the LT codes which are shown in Figure 4. The RUEP denotes the rateless unequal error protection (RUEP) code is which proposed to provide the different error protection property between the input symbols in the more important block and the less important block [16]. And the EWF codes is the Expanding Windowed Fountain (EWF) codes, in which codes are designed to use a simple method which is termed as expanding windowed to provide the UEP property [18]. To make the comparison fair, the LT, RUEP, and EWF codes are with the output degree distribution as shown in equation (15), and the RUEP and EWF codes are with the same UEP weight $K_M = 2$; it is worth noting that the UEP weight is also proposed by [16]. By observing Figure 4, it can be seen that the LT code with the lowest optimal decoding overhead at $\Gamma_{\mathrm{LT}} = 1.04$, which for RUEP and EWF codes are $\Gamma_{\mathrm{RUEP}} = 1.1$ and $\Gamma_{\mathrm{EWF}} = 1.54$, respectively. As for RUEP and EWF codes, the curves are the overall RRPS performances, which illustrate what the UEP property would provide at the price of lower transmission efficiency. And for the EWF codes, the simple method to provide UEP property would lead to a much lower transmission efficiency.

In the IoT of smart city scenarios, the small size of the data makes the traditional error correcting codes unable to provide good performances in such short length conditions; retransmission schemes based on the Automatic Repeat reQuest (ARQ) mechanism are widely adopted to provide the reliability transmission in the IoT and smart city scenarios. Because of that, the data size on each node in such scenarios is very limited; the transmission delay schemes would also lead to dramatic effects on the transmission efficiency of IoT and smart city scenarios. As the optimal decoding overhead Γ can be found, for simplicity, by considering a binary erasure channel with erasure probability E, then for the LT encoder, the optimal encoding overhead $\Gamma_{\mathrm{enc}} = \gamma/(1 - E)$ can be obtained. Then, the feedback messages of traditional LT schemes can be eliminated. It can be seen that by using the proposed LT scheme with optimal decoding overhead, the negative affections of transmission delay in IoT and smart city scenarios can be removed. We are given the comparison among the proposed LT and traditional LT schemes in Table 1, and the necessary parameters are given as follows. The data rate of the link is 50 kb/s, and the total transmission delay is 1 ms [6]; the channel is

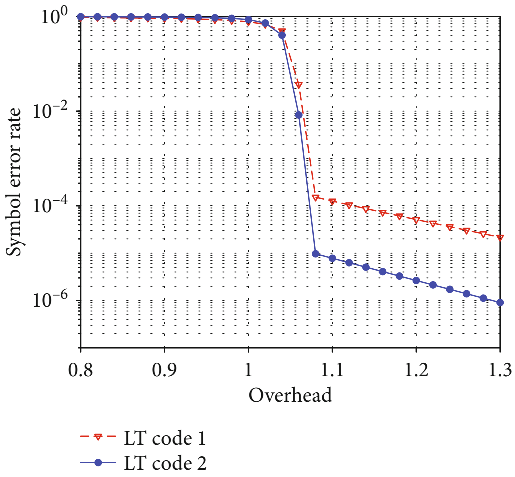

FIGURE 2: Symbol error rate performance of the LT codes with robust degree distributions.

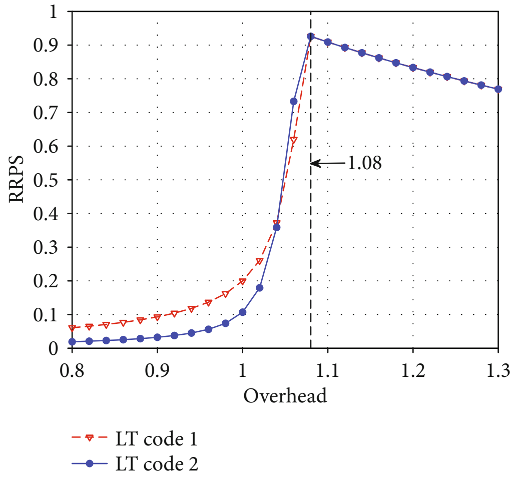

FIGURE 3: The overall RRPS performance of the LT and LT-based codes.

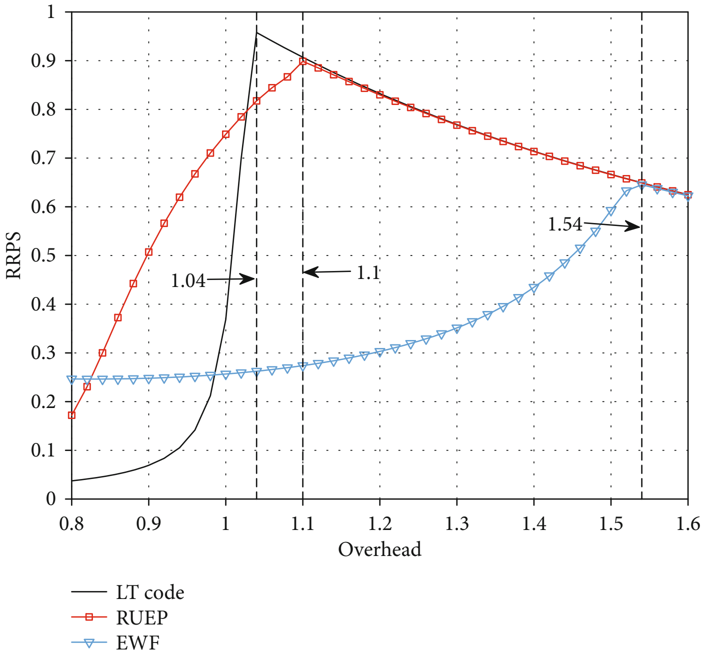

Figure 4: Asymptotic error performance of the given LT code.

Table 1: The transmission efficiencies of proposed and traditional LT schemes versus various target symbol error rates.

Target SER	10^{-1}	10^{-2}	10^{-3}	10^{-4}	10^{-5}
Scheme 1	93.75%	94.29%	91.74%	91.74%	90.91%
Scheme 2	85.23%	85.72%	83.40%	83.40%	82.65%
Scheme 3	81.82%	87.38%	76.39%	54.11%	46.38%

considered as a binary erasure channel and the data size is 1k bits. In the table, scheme 1 represents the proposed scheme, and the output degree distribution is redesigned based on each target symbol error rate. Scheme 2 adopted the traditional LT scheme, but the LT codes in this scheme share the same degree distributions as in scheme 1. The LT codes in scheme 3 have adopted the output degree distribution which is shown in equation (15), and the LT encoders are also terminated by adopting the traditional manner; specifically, the traditional manner to terminate the encoding process is the encoder that can receive a feedback message. By observing Table 1vv, one can find that scheme 1 provides higher transmission efficiencies than the other two schemes. Although schemes 1 and 2 share the same output degree distributions, affected by the feedback delay, the transmission efficiency of scheme 2 is obviously lower than that of scheme 1. As in scheme 3, the output degree distribution is not designed for the target symbol error rates, and the effect of feedback delay also exists, in which the scheme would have the worst transmission efficiencies.

6. Conclusion

By focusing on the nonnegligible negative effect of transmission delay in IoTs of a smart city, we proposed a novel LT scheme without feedback messages in this paper. By presenting the definitions of the recovery ratio per symbol and optimal decoding overhead, and given the necessary proof, the optimal decoding overheads of LT codes can be found. By using the channel estimate techniques in the smart city scenarios, the optimal encoding overhead can be obtained, then the feedback messages of traditional LT schemes can be removed. By utilising the proposed scheme, the transmission efficiency of IoTs of a smart city can be enhanced. We compared the symbol error rate, recovery rate, and recovery ratio per symbol performances of two given LT codes. The results showed that one can easily find out the optimal decoding overhead by using the recovery ratio per symbol, and the optimal encoding overhead can be found out as well. In a word, by using the proposed LT scheme, the nonnegligible negative effect of feedback delay in IoTs can be eliminated; thus, the proposed LT scheme can dramatically improve the transmission efficiency in the IoT of smart city scenarios.

Data Availability

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

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Authors' Contributions

Shuang Wu and Qingyang Guan contributed equally to this work.

Acknowledgments

This work was supported by the Scientific Research Initiation Funds for the Doctoral Program of Xi'an International University (Grant Nos. XAIU2018070102 and XAIU2019002), Regional Innovation Capability Guidance Project (Grant No. 2021QFY01-08), General Project of the Science and Technology Department of Shaanxi Province (Grant No. 2020JM-638), and Research Foundation of Education Bureau of Hunan Province (Grant No. 20B460).

References

[1] L. Sanchez, L. Munoz, J. A. Galache et al., "SmartSantander: Iot experimentation over a smart city testbed," *Computer Networks*, vol. 61, no. 6, pp. 217–238, 2014.

[2] M. Centenaro, L. Vangelista, A. Zanella, and M. Zorzi, "Long-range communications in unlicensed bands: the rising stars in the IoT and smart city scenarios," *IEEE Wireless Communications*, vol. 23, no. 5, pp. 60–67, 2016.

[3] J. Petjjrvi, M. Pettissalo, K. Mikhaylov, A. Roivainen, and T. Hnninen, "On the coverage of LPWANs: range evaluation and channel attenuation model for LoRA technology," in *2015 14th International Conference on ITS Telecommunications (ITST)*, Copenhagen, Denmark, December 2016.

[4] X. Liu and X. Zhang, "NOMA-based resource allocation for cluster-based cognitive industrial internet of things," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 8, pp. 5379–5388, 2020.

[5] F. Adelantado, X. Vilajosana, P. Tuset-Peiro, B. Martinez, J. Melia-Segui, and T. Watteyne, "Understanding the limits of LoRaWAN," *IEEE Communications Magazine*, vol. 55, no. 9, pp. 34–40, 2017.

[6] X. Liu and X. Zhang, "Rate and energy efficiency improvements for 5G-based IoT with simultaneous transfer," *IEEE Internet of Things Journal*, vol. 6, no. 4, pp. 5971–5980, 2019.

[7] X. Xiong, K. Zheng, R. Xu, W. Xiang, and P. Chatzimisios, "Low power wide area machine-to-machine networks: key techniques and prototype," *Communications Magazine IEEE*, vol. 53, no. 9, pp. 64–71, 2015.

[8] X. Liu, X. B. Zhai, W. Lu, and C. Wu, "QoS-guarantee resource allocation for multibeam satellite industrial internet of things with NOMA," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 3, pp. 2052–2061, 2021.

[9] M. Luby, "LT codes," in *The 43rd Annual IEEE Symposium on Foundations of Computer Science, 2002. Proceedings*, pp. 271–280, Vancouver, BC, Canada, 2002.

[10] A. Shokrollahi, "Raptor codes," *IEEE Transactions on Information Theory*, vol. 52, no. 6, pp. 2551–2567, 2006.

[11] J. Perry, H. Balakrishnan, and D. Shah, "Rateless spinal codes," in *Proceedings of the IEEE ACM HotNets*, pp. 6–11, New York, NY, USA, 2011.

[12] D. J. Costello and G. D. Forney, "Channel coding: the road to channel capacity," *Proceedings of the IEEE*, vol. 95, no. 6, pp. 1150–1177, 2007.

[13] I. F. Akyildiz, D. Pompili, and T. Melodia, "Underwater acoustic sensor networks: research challenges," *Ad Hoc Networks*, vol. 3, no. 3, pp. 257–279, 2005.

[14] M. M. Bennett and L. C. Smith, "Advances in using multitemporal night-time lights satellite imagery to detect, estimate, and monitor socioeconomic dynamics," *Remote Sensing of Environment*, vol. 192, pp. 176–197, 2017.

[15] M. G. Luby, M. Mitzenmacher, and A. Shokrallahi, "Analysis of random processes via And-Or tree evaluation," in *Proceedings of the Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 364–373, San Francisco, CA, USA, 1998.

[16] N. Rahnavard, B. Vellambi, and F. Fekri, "Rateless codes with unequal error protection property," *IEEE Transactions on Information Theory*, vol. 53, no. 4, pp. 1521–1532, 2007.

[17] A. Talari and N. Rahnavard, "On the intermediate symbol recovery rate of rateless codes," *IEEE Transactions on Communications*, vol. 60, no. 5, pp. 1237–1242, 2012.

[18] D. Sejdinovic, D. Vukobssratovic, A. Doufexi, V. Senk, and R. J. Piechocki, "Expanding window fountain codes for unequal error protection," *IEEE Transactions on Communications*, vol. 57, no. 9, pp. 2510–2516, 2009.