Zhang *et al. EURASIP Journal on Wireless Communications and Networking* (2020) 2020:116
https://doi.org/10.1186/s13638-020-01733-1

EURASIP Journal on Wireless Communications and Networking

RESEARCH Open Access

Research on correction algorithm of propagation error in wireless sensor network coding

Dongqiu Zhang[1†], Mingyong Pang[1*], Guangzhi Zhang[2†] and Dandan Huang[1]

* Correspondence: 895453915@qq.com
[†]Dongqiu Zhang and Guangzhi Zhang contributed equally to this work and considered co-first authors
[1]Education Science Department of Nanjing Normal University, Nanjing 210000, China
Full list of author information is available at the end of the article

Abstract

It is very difficult to deal with the problem of error correction in random network coding, especially when the number of errors is more than the min-cut of the network. We combine a small field with rank-metric codes to solve this problem in this paper. With a small finite field, original errors are compressed to propagated errors, and their number is smaller than the min-cut. Rank-metric codes are introduced to correct the propagated errors, while the minimum rank distance of the rank-metric code is hardly influenced by the small field. It is the first time to correct errors more than the min-cut in network coding with our method using a small field. This new error-correcting algorithm is very useful for the environment such as a wireless sensor network where network coding can be applied.

Keywords: Network coding, Error correction, Wireless sensor network, Small field

1 Introduction

The nature of combining information in intermediate nodes makes network coding very susceptible to transmission errors. How to control and correct the errors in random network coding is therefore of great interest to researchers. This problem naturally motivates the topic of network error correction (NEC), pioneered by Yeung and Cai [1]. Since then, a variety of models were presented to combat the problem. For a determined network, its topology is stable over time, Guang et al. [2–4] proposed several NEC construction algorithms for correcting propagated errors in it based on the Hamming metric. For random networks, changing their topologies over time, original errors can be spread to the down-stream nodes in the networks and cause the decoding failure when the Hamming metric is used. To solve the issue, Kotter [5] introduced a subspace/rank-metric-based method and developed an elegant approach. However, the capacity has reached the theoretical upper-bound of NEC ($C-2t$), where C and t are the min-cut of the network and the number of corrupted links in the network, respectively. That is to say, t , the number of errors to be corrected is no more than $C/2$, here $t = (C - R)/2$ and R is the transmission rate of the code. Guruswami et al. [6] generalized the rank-metric code to its list-decoding version. The method can correct nearly C errors when the transmission rate R is close to zero. In order to correct more errors, Guo et al. [7] introduced a nonlinear operation to network coding and proved that the transmission rate can be bigger than $(C - t)$ in her method. But she did not show her concrete construction and corresponding decoding algorithm about the method in the paper we also noticed. The operations in the method have exponential complexity. So far, there are few practical methods that can correct more than C errors in the related literature. However, in the real communication environments, the number of original errors is usually larger than min-cut C, since the number is directly proportional to total links in the networks.

In random network coding, the transport process can be depicted by $Y = T \cdot G \cdot u + T_{Z \to Y} \cdot \mathrm{Z}$, where u is the original message intended to be transmitted. In a source node of a multicast network, u is encoded to $G \cdot u$ by a code Ω with its generate matrix G. Then, $G \cdot u$ is sent to the network and encounters a transfer matrix T, which represents the effect of network coding upon $G \cdot u$. Simultaneously, the error Z occurred on the links also encounter their own transfer matrix $T_{Z \to Y}$, and then, $T_{Z \to Y} \cdot \mathrm{Z}$ are injected into the messages Y received in the sink node. In the existing NEC models, packets must be collected enough to guarantee that T is a full-rank matrix. By multiplying T^{-1} to both sides of the equation $Y = T \cdot G \cdot u + T_{Z \to Y} \cdot \mathrm{Z}$, we can obtain $T^{-1} \cdot Y = G \cdot u + T^{-1} \cdot T_{Z \to Y} \cdot \mathrm{Z}$. Based on the code Ω with its generated matrix G, we can further get the original message u. Denoted by Z, the original error, as the effect of network coding, there are several variations of Z such as $T^{-1} \cdot T_{Z \to Y} \cdot \mathrm{Z}$, $T_{Z \to Y} \cdot Z$, and they are called “propagated errors.” In essence, the existing NEC methods focused their aims on compressing the number of “propagated errors.” With respect to the determined network [2–4], the Hamming weight of the propagated error $T^{-1} \cdot T_{Z \to Y} \cdot Z$ is compressed to the Hamming weight of the original error Z. The spread effect of the original error Z is compressed by the NEC method. But for a random network, it is difficult to compress the spread of original error from the perspective of the Hamming metric. However, from the view of the rank metric, the rank of propagated error $T^{-1} \cdot T_{Z \to Y} \cdot Z$ is no more than the rank of the original error Z, and the spread of error in network coding is compressed naturally [5]. Once the list-decoding method is introduced, the range of the number of original errors that can be corrected increases from the interval $(0, C/2)$ to $(C/2, C)$, see [5] or [6] for a reference. In order to guarantee T is a full-rank matrix, the size of the finite field should be big enough [6]. The rank of propagated error $T^{-1} \cdot T_{Z \to Y} \cdot \mathrm{Z}$ could not be compressed to a smaller one than C when the rank of the original error Z is bigger than C. Even if the list decoding of rank-metric code [6] has a strong decoding ability, it still cannot correct the propagated errors based on the rationale of coding theory when the rank of propagated error $T^{-1} \cdot T_{Z \to Y} \cdot \mathrm{Z}$ is bigger than C.

In this paper, we propose a new method, which can correct more than C errors. In our experiments, we test and verify that if the size of a finite field for network coding is smaller than a threshold number, the rank of $T_{Z \to Y} \cdot Z$ can be smaller than C. But T unfortunately is usually not full rank, so we cannot use the decoding algorithm of code Ω with the generated matrix G to decode u, where the corresponding decoding equation is $T^{-1} \cdot Y = G \cdot u + T^{-1} \cdot T_{Z \to Y} \cdot Z$. For T is not full rank in a small field, we decode u based on the equation$Y = T \cdot G \cdot u + T_{Z \to Y} \cdot Z$. We will use the new code Ω’ with it generates matrix, $T \cdot G$, instead of G to decode u. Here, the rank of $T_{Z \to Y} \cdot Z$ is smaller than C. Now, the only question remaining is how much the minimum rank distance of code Ω' declines, compared to the minimum rank distance of code Ω. If the minimum distance of code Ω' declines too much, we still cannot correct the propagated error $T_{Z\rightarrow Y}\cdot Z$ with the list-decoding algorithm of rank-metric code, just like that in [7]. Fortunately, based on our experiments to be shown in Section 5, we found that the minimum distance of $T \cdot G$ is very close to the minimum distance of G even if T is not full rank when a small field is used. In the traditional research paradigm of network coding, we use a big field to make sure T is full rank. In our work, we deal with the problem by using a small field way to compress the spread of original error based on the rank metric. Even the number of non-zero components in original error Z is far larger than C, the rank of propagated error $T_{Z\rightarrow Y}\cdot Z$ also can be compressed to a smaller number than C as long as the size of the field is small enough. Our method not only can correct more than C errors in random network coding, but also can correct lots of errors valuable to practical applications of network coding.

The rest of this paper is organized as follows. We first introduce the related works including rank-metric codes and list decoding briefly in Section 2. Then, we present our method in Section 3. In Section 4, we give the experiment results and make a detailed theoretical analysis based on a combinatorial theory of probabilities and simulation. Finally, we draw a conclusion about our work in Section 5.

2 Related works

In this section, as the background of our work, we first introduce the rank-metric codes and the list-decoding technique. Then, we explain a model about NEC, in which list decoding of rank-metric codes is involved.

2.1 Rank-metric codes

Denote the set of all $n \times m$ matrices over F_q by $F_q^{n\times m}$ and suppose a rank-metric code, Ω, is a subset of $F_q^{n\times m}$. Define the distance between $X\in F_q^{n\times m}$ and $Y\in F_q^{n\times m}$ as rank(X − Y). Obviously, we can also take the matrix $X\in F_q^{n\times m}$ as a vector $x\in(F_{q^m})^n$ in the field F_{q^m}, and this means there is a bijection between the vector set of F_{q^m} and the matrix set of $F_q^{n\times m}$. We further define rank(x) = rank(X).

2.2 List decoding of rank-metric codes

In the traditional decoding method of linear block codes, the solution is unique, and the number of errors that can be corrected is less than the half-size of the codeword. With the method of list decoding, we can correct errors more than half of a codeword size, but the solutions of list-decoding method are not unique. Based on the rank codes, Guruswami in [6] proposed the list decoding of rank codes and approximate the number of the corrected errors to the word size when the transmission rate approximates zero. We can utilize the method with its strong error-correcting ability to correct the propagated error in network coding, where the codeword size is exactly the min-cut C.

2.3 Graph model about network coding

Consider an acyclic directed graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, where $\mathcal{V}$ is a node set and, $\mathcal{E}$ is an edge set whose elements represent network channels. A channel $e = (i, j)$ is a directed edge starting from node i and ending at node j, i.e., tail(e) = i and head(e) = j. For a node$i$, the collection of incoming channels is In(i) $\{e : e \in \mathcal{E}, \text{head}(e) = i\}$ and the collection of outgoing channels is out(i) = $\{e : e \in \mathcal{E}, \text{tail}(e) = i\}$. Each channel has a unit capacity in the network. NEC is specified by the related encoder at the source, encoders at intermediate nodes, and decoders at sink nodes. The coding and decoding operations for messages are performed over the field F_{q^m}, and the network coding is F_q.

3 Proposed methods

3.1 Some claims based on numerical experiments

In this section, we give a set of numerical experiment results to be used in our method. The results are summed up in our three claims. The experiment details will be given in the next section. The three claims establish the base of our method in Section 3. Our claims are based on plenty of experiments rather than rigorous mathematical deducing. However, we also give out auxiliary mathematical analysis about the claims in Section 4.

Claim 1:

Suppose $|F_q| \in \{2, 3, 4, 5\}$, there is $\text{rank}(T_{Z \to Y} \cdot Z') < C$ with a high probability, where $Z' \in (F_q)^{t \times m}$, $T_{Z \to Y} \in (F_q)^{C \times t}$, and $t > 0$.

Claim 2:

For rank-metric code Ω with the generated matrix G and minimum rank distance $d_{\min}$, if $\text{rank}(T) = C - k$, then $d_{\min}' = d_{\min} - k$ is the minimum rank distance $d_{\min}'$ of code Ω’ with the generated matrix $T \cdot G$, where $T \in (F_q)^{C \times C}$, $0 < k < C$.

Claim 3:

If $|F_q| \in \{2, 3, 4, 5\}$, we have $\text{rank}(T) = C - 1$, where $T \in (F_q)^{C \times C}$ with high probability.

3.2 Data collection strategy with delay constraint

In this section, we formally propose our method. The method is specified by its encoder at the source, encoders at intermediate nodes, and decoders at sink nodes. The coding operation is over the field F_{q}.

3.2.1 Source

A source message $u \in (F_{q^m})^R$ is encoded with a rank-metric code Ω equipped with the generated matrix $G \in (F_{q^m})^{C \times R}$, where $m \geq C$ and $0 < R < C$. The coded message is a vector $x \in (F_{q^m})^C = G \cdot u$. Then, x is then sent to the network from the source. The minimum rank distance of Ω is $d_{\min}$.

3.2.2 Coding at intermediate nodes

Every intermediate node combines its received packets from incoming edges with its own local coding kernel in field F_q, creates a new packet, and sends the packet to its successors via outgoing edges. At a more macroscopic level, the network coding at that time leads that messages encounter a so-called transfer matrix [8]. In our method, the messages $x \in (F_{q^m})^C = G \cdot u$ encounter a transfer matrix $T \in (F_q)^{C \times C}$ and the error $Z \in (F_{q^m})^t$ encounters a corresponding transfer matrix $T_{Z \to Y} \in (F_q)^{C \times t}$, where t is the number of edges, on which errors occur. Roughly speaking, the received messages Y in the sink can be expressed as $Y = T \cdot G \cdot u + T_{Z \rightarrow Y} \cdot Z$, if T is not polluted by errors. T is gotten from the global coding kernels in the packets. However, if errors occur, the global coding kernels may also be polluted. At this case, T is then naturally polluted and it is unknown. Though T is unknown, the polluted version of it, denoted by $\hat{T}$, is known. The transmission procedure can be expressed by a more delicate equation $Y = \widehat{T \cdot} G \cdot u + T_{Z \rightarrow Y} \cdot (Z - L \cdot G \cdot u)$. Here, $L \in (F_q)^{t \times C}$ is a matrix, formed by grouping the global coding kernel vectors together.

3.2.3 Decoding in the sink

Let $\hat{T} \cdot G$ as the generated matrix of the newly formed rank-metric code Ω' with a minimum rank distance of $d_{\min}'$ and Y as the received messages, and the errors can be evaluated as $T_{Z \rightarrow Y} \cdot (Z - L \cdot G \cdot u)$. We utilize the list-decoding method of rank-metric codes [6] to perform decoding and get u, as long as $\operatorname{rank}(T_{Z \rightarrow Y} \cdot (Z - L \cdot G \cdot u)) < C$ and $d_{\min}'$ is not smaller than $d_{\min}$ too much. This means u^d, corresponding to $Y^d = \underset{Y^d \in \Omega'}{\arg} \min \operatorname{rank}(Y^d - Y)$, is the estimate of u. So, u^d is the solution of the decoding algorithm for the original *message*. In the list decoding, u^d is not unique.

3.2.4 The feasibility of the decoding

We discuss the feasibility of our method here. The feasibility depends on claims 1 to 3. As mentioned in Section 2, the model in [6] to correct the errors is based on the equation $\hat{T}^{-1} \cdot Y = G \cdot u + \hat{T}^{-1} \cdot T_{Z \rightarrow Y} \cdot (Z - L \cdot G \cdot u)$ in the random network coding. $\hat{T}^{-1} \cdot Y = G \cdot u + \hat{T}^{-1} \cdot T_{Z \rightarrow Y} \cdot (Z - L \cdot G \cdot u)$ is induced by multiplying $\hat{T}^{-1}$ on both sides of the equation $Y = \widehat{T \cdot} G \cdot u + T_{Z \rightarrow Y} \cdot (Z - L \cdot G \cdot u)$, and $\hat{T}^{-1}$ is invertible in the field F_q. We use the decoding algorithm of code Ω with its generated matrix G to perform list decoding. The list-decoding produce can correctly work as long as $\operatorname{rank}(\hat{T}^{-1} \cdot T_{Z \rightarrow Y} \cdot (Z - L \cdot G \cdot u)) < C$. It is a dilemma to make a choice, using a bigger field size or using a smaller field size in the context of random network coding. If $\hat{T}^{-1}$ is invertible, the size of field F_q would be big enough, and it is usually $|F_q| \geq 256$ [9]. On the other hand, claim 1 shows $\operatorname{rank}(\hat{T}^{-1} \cdot T_{Z \rightarrow Y} \cdot (Z - L \cdot G \cdot u)) > C$ if $|F_q|$ is bigger than 5. In this case, the decoding certainly fails if the list-decoding method [6] is used.

As discussed above, we use code Ω' with the generated matrix $\widehat{T \cdot} G$ to perform list decoding, where the corresponding error is $T_{Z \rightarrow Y} \cdot (Z - L \cdot G \cdot u)$ rather than $\hat{T}^{-1} \cdot T_{Z \rightarrow Y} \cdot (Z - L \cdot G \cdot u)$. The two preconditions for successfully decoding are that $\operatorname{rank}(T_{Z \rightarrow Y} \cdot (Z - L \cdot G \cdot u)) < C$ and $d_{\min}'$ is not smaller than $d_{\min}$ too much, respectively. The first condition, $\operatorname{rank}(T_{Z \rightarrow Y} \cdot (Z - L \cdot G \cdot u)) < C$, can be naturally satisfied, and it is necessary according to the inherent nature of the linear block code. For the second condition, if $d_{\min}'$ is not smaller than $d_{\min}$ too much, we can consider that the code Ω' has a nearly identical error-correcting ability with the code Ω. When $\operatorname{rank}(T_{Z \rightarrow Y} \cdot (Z - L \cdot G \cdot u)) < C$, we also perform list decoding of rank-metric just like done in [6]. If $d_{\min}'$ is smaller than $d_{\min}$ too much, the error-correcting ability of code Ω' declines sharply, compared with code Ω. In this case, it becomes un-meaningful in practice, even if code Ω' can correct nearly C errors in theory. After the theoretical analysis, we explain that the two preconditions can be met in the following case if $|F_q| \in \{2, 3, 4, 5\}$. In order to obtain better results, we set $|F_q| = 2$ in this paper.

Based on claim 1, $\text{rank}(T_{Z\rightarrow Y} \cdot (Z - L \cdot G \cdot u))$ is smaller than C if $|F_q| = 2$. In essence, $\text{rank}(T_{Z\rightarrow Y} \cdot (Z - L \cdot G \cdot u)) \leq \text{rank}(T_{Z\rightarrow Y} \cdot Z)$ because $rank(Z - L \cdot G \cdot u) \leq rank\,(Z)$.

Based on claims 2 and 3, it can be guaranteed that $d_{\min}'$ is not smaller than $d_{\min}$ too much. In most cases, $d_{\min}' = d_{\min}$ or $d_{\min}' = d_{\min} - 1$, and in a few cases, $d_{\min}' = d_{\min} - 2$. The specific situation depends on the sizes of C and m. The details of it will be introduced in the experiment section.

3.2.5 Advantages and disadvantages

The advantages of our method are as follows: (1) More than C errors can be corrected with list decoding of rank-metric codes in a random network coding, when we adopt a small network coding field, for example, $|F_q| = 2$. (2) Our method has the ability of correcting the original more than min-cut C errors, and it is very important in the real applications of network coding. (3) The small field can avoid a low computational burden. In the rank-metric code for network coding, we can make the extension field F_{q^m} big enough to get a bigger $d_{\min}$. In this case, m is usually bigger than min-cut C [10]. In [9], $|F_q|$ is set more than 256 to guarantee $\hat{T}$ is invertible. Naturally, a bigger F_{q^m} obviously lead to a heavy computation. In our approach with a small field, F_{q^m} should be smaller because $|F_q|$ is very small, and a small F_{q^m} can lead to a serious computational burden. (4) The disadvantage of our approach is that the transmission rate is a little smaller than the rate in [6] because $d_{\min}'$ is 1 or 2 and it is smaller than $d_{\min}$. But the problem can be alleviated as C becomes bigger because 1 or 2 is a few ratios of C. In this case, $d_{\min}'$ is also sufficient to successfully decode. On the other hand, the more components about the original error Z, i.e., t, the bigger the rank of the propagated errors $T_{Z\rightarrow Y} \cdot (Z - L \cdot G \cdot u)$ it takes. In this case, the transmission rate is usually low.

4 Experimental results and discussion

In this section, we give a set of experiments to support claims 1 to 3 in Section 3. We also give corresponding theory analysis about experiments. Because many mathematical operations are performed in the finite field, we design our program about the finite field mathematical operation based on MATLAB, such as inversing a matrix, computing the inverse of a number in the finite field, and computing rank of a matrix in the finite field. In the experiments, we use newly designed finite operation methods to verify claims 1 to 3.

4.1 Experimental results

4.1.1 Claim 1

Assume the min-cut $C = 5 * |F_q|$.

In Fig. 1, the different numbers of original errors are illustrated by different curves. We can find that, if $|F_q| = 2$, the rank of the propagated errors $T_{Z\rightarrow Y} \cdot (Z - L \cdot G \cdot u)$ can be compressed to $0.7 * C$, even if $t = 20 * C$, where t is the number of original errors Z. In this case, the list decoding of rank-metric codes can correct the propagated errors $T_{Z\rightarrow Y} \cdot (Z - L \cdot G \cdot u)$ easily. This character is good for correcting the dense errors in the random network coding.

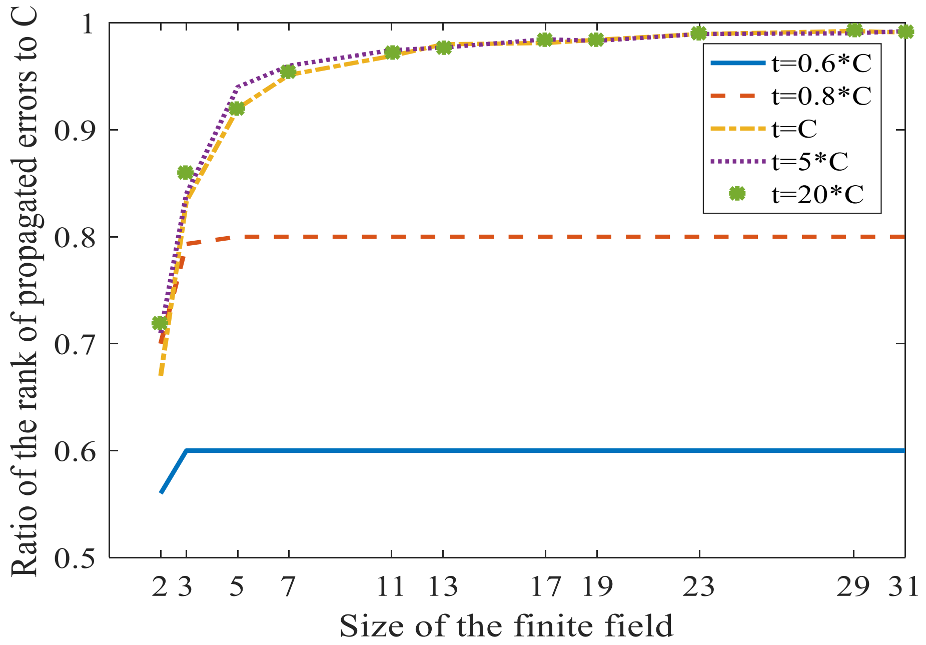

Fig. 1 Normalized rank of propagated error $T_{Z\to Y}\cdot(Z-L\cdot G\cdot u)$ based on claim 1 when the size of finite field is odd. Legend: The vertical axis value is $\frac{\text{rank}(T_{Z\to Y}\cdot(Z-L\cdot G\cdot u))}{C}$. A smaller field size implies a smaller value about normalized rank of propagated error, when *t* (the number of original error) is fixed

Because it is difficult to do finite field programming with MATLAB when the size of the finite field is even, we use C programming language to do finite field programming. Like Fig. 1, where the size of the finite field is odd, Fig. 2 shows how the errors are propagated in the finite field when the size of the finite field is even.

We now analyze the experimental results theoretically and find that there is a satisfactory coincidence with the theory and the experiment results. The transfer matrix $T_{Z\to Y}\in(F_q)^{C\times t}$ is known in advance, where t is the number of edges on which errors occur, and $(Z-L\cdot G\cdot u)$ takes its value in $(F_{q^m})^t$. Based on the theory of the extension field and the base field, we can take the vector $(Z-L\cdot G\cdot u)$ in field F_{q^m} as a matrix $(F_q)^{t\times m}$ in field F_q [10]. $T_{Z\to Y}\cdot(Z-L\cdot G\cdot u)$ means that when we multiply two matrices together in the field F_q, they should satisfy $(C\cdot t)\times(t\cdot m)=(C\times m)$. Finally, we can assert $T_{Z\to Y}\cdot(Z-L\cdot G\cdot u)\in(F_q)^{C\times m}$. Obviously, no matter how big the value t is, the rank of $T_{Z\to Y}\cdot(Z-L\cdot G\cdot u)$ is no more than C. The reason is that a small field F_q usually makes rank$(T_{Z\to Y}\cdot(Z-L\cdot G\cdot u))\le C$, no matter how big the value of t is. If the Hamming metric is adopted, no matter how small the size of F_q is, we cannot compress Hamming weight of $T_{Z\to Y}\cdot(Z-L\cdot G\cdot u)$ to the number smaller than C. Naturally, we cannot correct the propagated errors even though the list decoding is used based on the Hamming metric.

4.1.2 Claim 2

According to the experiments we have done, we found that claim 2 always holds on all the parameter combinations. So far, no counterexample has been found to claim 2 in our work.

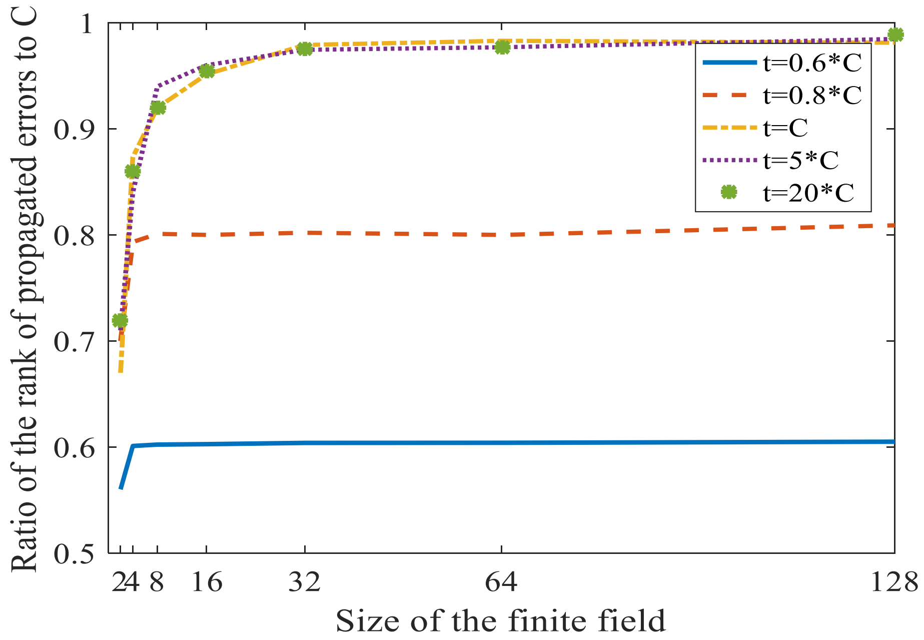

Fig. 2 Normalized rank of propagated error $T_{Z\rightarrow Y}\cdot(Z-L\cdot G\cdot u)$ based on claim 1 when the size of a finite field is even. Legend: The vertical axis value is $\frac{\mathrm{rank}(T_{Z\rightarrow Y}\cdot(Z-L\cdot G\cdot u))}{C}$. A smaller field size implies a smaller value about normalized rank of propagated error, when t (the number of original error) is fixed

4.1.3 Claim 3

In Table 1, we show several examples about the decline of the rank of T. We can see the declined amount is 0, 1, or 2.

In Fig. 3, we illustrated the ratio of the declined amount of the rank of T in a small field to the full rank. We can find the ratio very low, which means $d_{\min}$' is very close to $d_{\min}$ when the size of the finite field is odd.

Because it is difficult to do finite field programming with MATLAB when the size of the finite field is even, we use C programming language to do finite field programming. Like Fig. 3 where the size of the finite field is odd, Fig. 4 shows how the rank reduced in the finite field when the size of the finite field is even (Table 1).

Whether T is a full rank depends on the size of the finite field. Figure 5 shows the probability that $T\in(F_2)^{C\times C}$ is a full rank in field F_2.

5 Discussion

Consider the rank of a square matrix T in the finite field F_q. The probability that $T\in(F_q)^{C\times C}$ is a full rank in field F_q is $(1-|F_q|^{-C})\times(1-|F_q|^{-(C-1)})\times\cdots(1 - |F_q|^{-2})\times(1 - |F_q|^{-1})$ [9]. For the first selected row of T, the probability that this row is a nonzero vector is $1-|F_q|^{-C}$. For the second selected row of T, the probability this row is linearly

Table 1 Several examples about the decline of T. For example, the rank of T is 3 ($C-2$) when $C=5$ and the field size is 2. In this case, the declined amount about the rank of T is 2.

Field size	2	3	5	7	11
C = 5	2	0	0	0	0
C = 20	1	1	0	0	0
C = 50	1	2	1	0	0

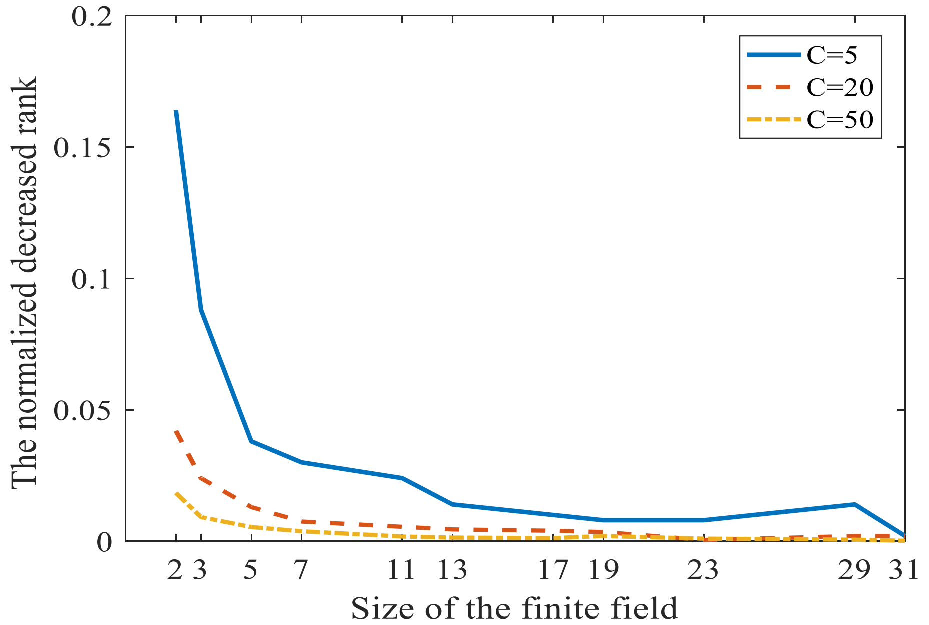

Fig. 3 Normalized decreased rank based on claim 3 when the size of a finite field is odd. Legend: Denote declined amount about rank of T is by ∂, and the vertical axis characters are ∂/C. In this case, matrix dimension T is $C \times C$, and its rank is $C - \partial$

independent with the first selected row is $1-|F_q|^{-(C-1)}$. The rest rows can also be considered in a similar way. Consider the case that $|F_q| = 2$ and C approximate infinity, the probability that T is a full rank is about 0.289 based on [9]. The probability rank of T that declines no more than 2 is $(1-2^{-C}) \times (1-2^{-(C-1)}) \times \cdots (1-2^{-3})$, where $(1-2^{-2}) \times (1-2^{-1})$ is not included. If C approximates infinity, $(1-2^{-C}) \times (1-2^{-(C-1)}) \times$

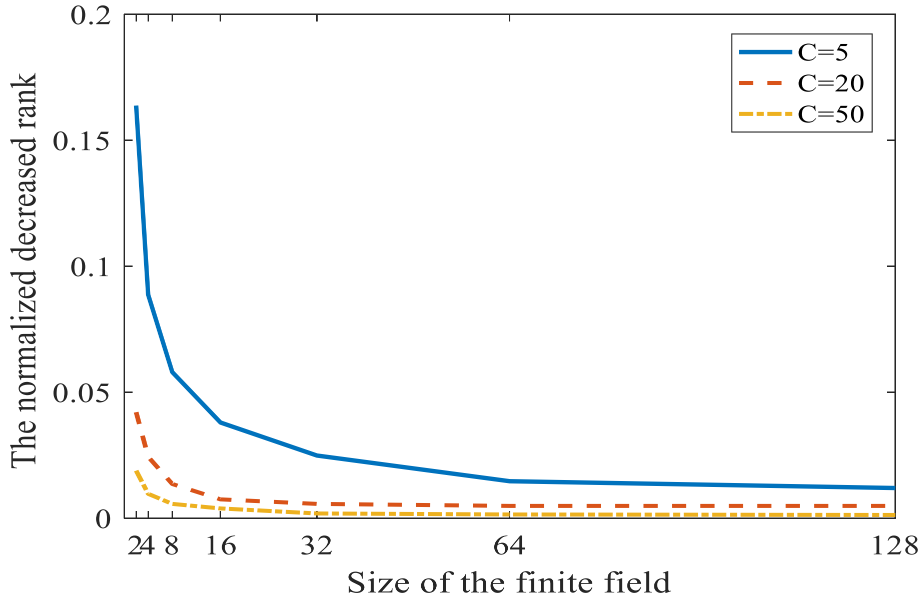

Fig. 4 Normalized decreased rank based on claim 3 when the size of a finite field is even. Legend: Denote declined amount about rank of T is by ∂ and vertical axis characters are ∂/C. In this case, matrix dimension T is $C \times C$, and its rank is $C - \partial$

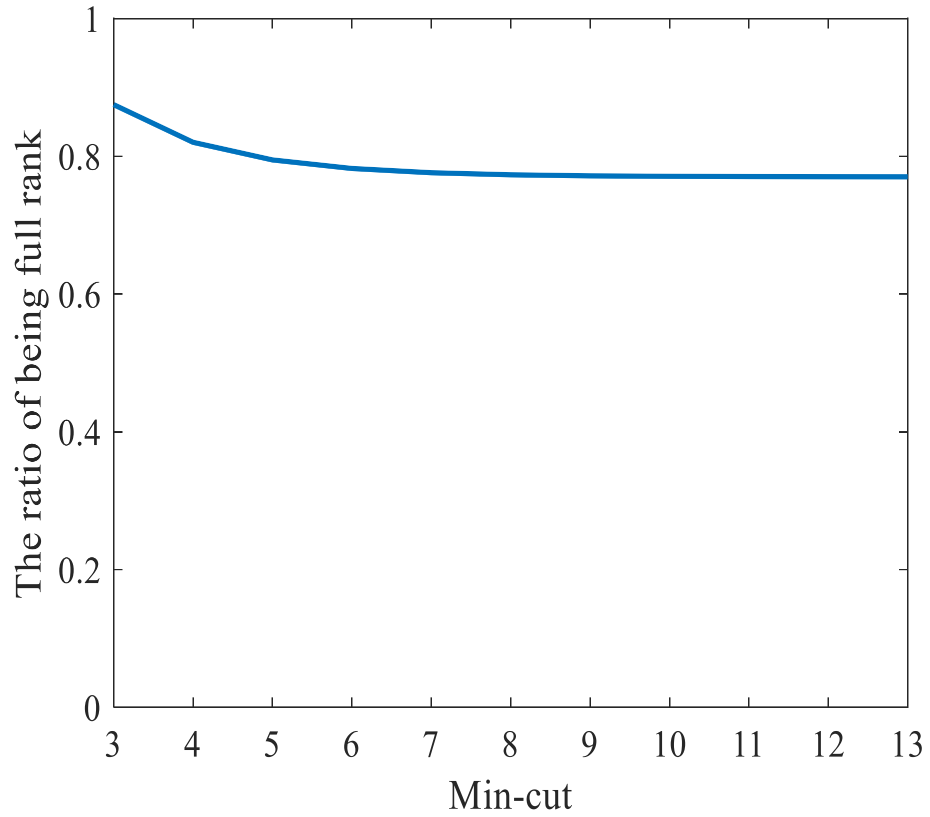

Fig. 5 The probability that $T \in (F_2)^{C \times C}$ is a full rank. Legend: Only the finite field F_2 is considered

$\cdots(1 - 2^{-3})$ is about 0.7707. In Fig. 2, when $|F_q| = 2$ and $C = 50$, the normalized declined amount about rank is very low. So, the rank of T also did not decline too much when $|F_q| = 2$, and then, it means $d_{\min}'$ is very close to $d_{\min}$.

This work mainly depends on information methods, if we adopt the methods from the machine learning field [11–14], we may correct more errors. The methods in [15–18] are also worth learning for the network coding error correction.

6 Conclusions and future work

With a small field $|F_q| = 2$, the number of original errors can be compressed to less than the min-cut of the network when their number is far more than the min-cut of the network. The minimum distance of the newly formed rank-metric code in the small field also did not decline sharply. So, the propagated errors can be corrected by the list-decoding method of the newly formed rank-metric code. Our scheme can correct more than the min-cut errors in network coding. A small field is also useful to reduce the computational burden compared to the bigger fields.

In our future research, we will try to optimize the scheme in real scenarios. We also will combine deep learning methods with network coding to improve the effectiveness of this method.

Abbreviations

NEC: Network error correction

Acknowledgements
The authors acknowledged the anonymous reviewers and editors for their efforts in valuable comments and suggestions.

Authors' contributions
Dongqiu Zhang and Guangzhi Zhang conceived the idea almost at the same time when they are discussing the problem. Guangzhi Zhang designed the algorithm of the experiment. Mingyong Pang organized the work and wrote the paper. Dandan Huang assisted in the laboratory work and analyzed the results. The authors read and approved the final manuscript.

Authors' information
Dongqiu Zhang is pursuing a doctorate degree in the Education Science Department of Nanjing Normal University. Her primary research interests include network coding, deep learning, and education technology.
Mingyong Pang was born in 1968, Ph.D., a professor in the Computer Department, Nanjing Normal University. Post doctor in the Computer Department, Nanjing University. His primary research interests include digital set processing and geometry-based modeling and rendering.
Guangzhi Zhang was born in 1982, Ph.D. in Computer Science, and an associate professor of Suihua University, China. His primary research interests include ad hoc networks and deep learning.
Dandan Huang is pursuing a master's degree in the Education Science Department of Nanjing Normal University. Her primary research interests include digital image processing.

Funding
This work was supported in part by Natural Science Foundation of Heilongjiang Province under Grant LH2019F052, and in part by Key topics of the 13th five year plan for Education Science in Heilongjiang Province in 2019 under Grant GJB1319161, in part by Key topics of the Ministry of education in 2017 in the 13th five year plan of National Education Science under Grant DCA170302, in part by Key Project of Natural Science Foundation of China under Grant 41631175, Innovation and entrepreneurship project of college students in Heilongjiang Province under Grant 201810236051.

Availability of data and materials
Data sharing is not applicable to this article as no datasets were generated or analyzed during the current study.

Competing interests
The authors declare that they have no competing interests.

Author details
[1]Education Science Department of Nanjing Normal University, Nanjing 210000, China. [2]Information Engineering Department of Suihua University, Suihua 152000, China.

Received: 16 February 2020 Accepted: 14 May 2020
Published online: 05 June 2020

References

1. R.W. Yeung, N. Cai, Network error correction, I: Basic concepts and upper bounds. Commun. Inf. Syst. **6**(1), 19–31 (2006)
2. G. Xuan, F.W. Fu, Z. Zhang, in *International Symposium on Networking Coding*. Construction of network error correction codes in packet networks (2011), pp. 1–6
3. Z. Zhang, Linear network error correction codes in packet networks. IEEE Trans Inform Theory **54**, 209–218 (2008)
4. S. Yang, R.W. Yeung, K.N. Chi, Refined coding bounds and code constructions for coherent network error correction. IEEE Trans Inform Theory **57**, 1409–1424 (2010)
5. R. Koetter, F.R. Kschischang, Coding for errors and erasures in random network coding. IEEE Trans Inform Theory **54**, 3579–3591 (2007)
6. V. Guruswami, C. Wang, C. Xing, Explicit list-decodable rank-metric and subspace codes via subspace designs. IEEE Trans Inform Theory **62**, 2707–2718 (2016)
7. W. Guo, H. Dan, C. Ning, On capacity of network error correction coding with random errors. IEEE Commun. Lett. **22**, 696–699 (2018)
8. R. Koetter, M. Medard, in *Joint Conference of the IEEE Computer and Communications Societies*. Beyond routing: an algebraic approach to network coding (Proceedings, IEEE, 2002), pp. 122–130
9. P.A. Chou, *Practical network coding. Allerton Conference on Communication, Control, and Computing, Monticello* (2003)
10. È. Gabidulin, Theory of codes with maximum rank distance. Probl. Inf. Transm. **21**, 3–16 (1985)
11. W.T. Cheng, Y. Sun, G.F. Li, G.Z. Jiang, H.H. Liu, Jointly network: A network based on CNN and RBM for gesture recognition. Neural Comput. Applic. **31**(Suppl 1), 309–323 (2019)
12. G.F. Li, D. Jiang, Y.L. Zhou, G.Z. Jiang, J.Y. Kong, G. Manogaran, Human lesion detection method based on image information and brain signal. IEEE Access **7**, 11533–11542 (2019)
13. G.F. Li, L.L. Zhang, Y. Sun, J.Y. Kong, Towards the sEMG hand: Internet of things sensors and haptic feedback application. Multimed. Tools Appl. **78**(21), 29765–29782 (2019)
14. J. X Qi, G.Z Jiang, G.F Li, Y. Sun, B. Tao, Intelligent human-computer interaction based on surface EMG gesture recognition, IEEE Access, 2019, 7: 61378-61387.
15. Z. Huang, X. Xu, J. Ni, H. Zhu, W. Cheng, Multimodal representation learning for recommendation in internet of things. IEEE Internet Things J. **6**(6), 10675–10685 (2019)
16. T. Zhou, J. Zhang, Analysis of commercial truck drivers' potentially dangerous driving behaviors based on 11-month digital tachograph data and multilevel modeling approach. Accid. Anal. Prev. **132**, 105256 (2019)

17. Z. Chen, Y. Zhang, C. Wu, B. Ran, Understanding individualization driving states via latent Dirichlet allocation model. IEEE Intell. Transp. Syst. Mag. **11**(2), 41–53 (2019)
18. Z. Huang, J. Tang, G. Shan, J. Ni, Y. Chen, C. Wang, An efficient passenger-hunting recommendation framework with multi-task deep learning. IEEE Internet Things J. (2019). https://doi.org/10.1109/JIOT.2019.2901759

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.