RTDCM: A Coding Preemption Collection System for Key Data Prioritization With Hierarchical Probability Exchange Mechanism in Mobile Computing

WEI ZHANG, DANLONG CHEN, HUAYOU SI, AND NEAL N. XIONG

1Department of Computer Science and Technology, Hangzhou Dianzi University, Hangzhou 310037, China
2Department of Mathematics and Computer Science, Northeastern State University, Tahlequah, OK 74464, USA

Corresponding author: Huayou Si (sihy@hdu.edu.cn)

This work was supported by the National Natural Science Foundation of China under Grant 61772165.

ABSTRACT Mobile computing technology will enable computers or other information intelligent terminal devices to enable data transmission and resource sharing in a wireless environment. In industrial or disaster situations, the destruction of sensor nodes is unpredictable, and some urgent data, such as information about damage points in industrial equipment, needs to be immediately transmitted to decision makers. Traditional data collection protocols, such as Growth Codes, cannot distinguish the importance of data. This behavior increases the risk of losing important data. In this article, we analyze the impact of the Growth Codes on the “fair” processing of all data. From the perspective of distinguishing the importance of data, a real-time data collection model (RTDCM) is proposed. This model enables important data to capture the coding opportunities of common data, increase the proportion of code words with important data in the overall code word of the network, and layer the network, so that the data transmission has a certain purpose, and ultimately improve the decoding efficiency of important data. According to different preemption methods, two RTDCM-based protocols are designed. Experiments show that the data recovery characteristics of RTDCM-based data acquisition protocols are mainly concentrated on important data, and have high recovery efficiency for ordinary data.

INDEX TERMS Data collection, growth codes, intelligent algorithm, mobile computing, network coding, wireless sensor network.

I. INTRODUCTION

The Mobile computing [26] is a new type of computing model spawned by wireless communications, computer technology and portable information devices, in which users can get the data and services they need, no matter how their physical location moves. Wireless sensor network is an important branch of the Mobile computing technology. It is small and convenient, and it undertakes the task of collecting information in the industrial environment. Sensor nodes have limited computing power and storage capacity and are in extreme environments. The basic purpose of a sensor network is to monitor an emergency or disaster situation.

In recent years, it has also been applied to slope monitoring [1], health care [2], and related applications in the industrial field [24].

The introduction of network coding [5] has greatly improved the reliability of data in sensor networks. Among them, the representative research work is Growth Codes scheme proposed by Kamra et al. [3], which has good performance in harsh environments. Growth Codes can generate a large number of source data backups in the network through a well-designed codeword conversion sequence and a randomly distributed data replication method, thereby improving the reliability of the data in the network and well solving data transfer problem in the zero-configuration network. However, Growth Codes cannot distinguish the importance of the data, but is freely encoded and transmitted along with the ordinary...
data. This can cause delays in important data and increase the risk of losing important data.

In view of the above problems, this paper proposes a real-time data collection model based on the Growth Codes scheme. Based on this model, two types of data collection protocols are proposed. Important data can quickly reach the sink node, which will increase the collection efficiency and data reliability of important data in the wireless peer-to-peer network. Our work includes the following sections:

1. The layering mechanism is used to make the node roughly understand the location of the aggregation node, and send the codeword to the node of the next level with a high probability, thereby reducing the number of invalid transmissions of the codeword and improving the data recovery efficiency of the sink node.

2. A priority-based preemptive coded data collection model is proposed. When a node encodes data, a codeword containing important data has a higher encoding opportunity than a ordinary codeword. This can increase the proportion of codewords containing important data in the overall codeword. Based on this model, we have designed two different preemption methods to deal with different application scenarios.

3. Growth Codes uses a random exchange policy in the node data exchange strategy, which may result in codewords containing more important data being exchanged away from the sink node. Therefore, we design a probability forwarding table using neural network algorithms and genetic algorithms. Each time the node forwards the codeword, it exchanges the codeword with the neighbor with different probability by querying the probability forwarding table according to the important data in the codeword. Codewords with more important data will most likely be sent near the sink node. Sink nodes can obtain higher value codewords, which improves the recovery efficiency of important data.

In the II chapter we will introduce related works, the III chapter will introduce the system model and some definitions, the IV chapter describes the model framework and the model-based two protocols in detail, the V chapter will analyze the two protocols Performance, summary and outlook in chapter VI.

II. RELATED WORKS

A. NETWORK CODING

It is well known that by using network coding technology, the data persistence of the sensor network can be significantly improved, and the throughput of the network is increased, which allows intermediate nodes to combine data received from different links [6]. In this case, the network has a certain routing function, and each node knows the number and maximum capacity of its input and output links. Katie et al. [5] proposed the COPE coding, and established the basis of wireless sensor network coding research based on XOR operation. Because there are too many nodes trying to pass data to the receiver, there is congestion and delay near the receiver, which can be described as a funnel effect [4].

B. DELAYED DATA COLLECTION PROTOCOL

In the aspect of delayed data collection protocol, the coding technology represented by digital fountain code [7] is widely used in data collection and storage of wireless sensor networks, but it just stays at the theoretical level. Nguyen et al. [13] studied the advantages of using fountain code (FC) to improve the transmission efficiency of broadcast systems. Luby [8] makes the fountain code work in practice. LT Code [8] is a rateless linear random fountain code. The decoding algorithm is simple and efficient, but it can’t be decoded at a fixed cost. LT Code greatly reduces the encoding. The complexity extends the range of applications for network coding. Based on LT Code, Shokrollahi. Reference [9] proposed an improved LT code with precoding mechanism called Raptor code [9], which uses a two-step encoding mechanism and uses LDPC [10] encoding.

Lin et al. [11] proposed a new dispersion algorithm based on fountain code in large-scale sensor networks, which can make the data completely dispersed by using the low decoding complexity of the fountain code and the scalability of the random walk propagation process. Talari and Rahna [12] proposed a robust LT code with feedback function, designed a parameterless coding degree distribution for LT-AF coding, and improved decoding efficiency. Ostovari and Wu [18] used random linear coding for distributed storage systems to store large amounts of data on different storage and provide fault tolerance for storage failures.

C. FAST DATA COLLECTION PROTOCOL

In terms of real-time data collection protocols, reference [14] proposed a wireless sensor network data collection protocol based on the theory of compressed sensing [15], which improves the data collection efficiency of the network and prolongs the life cycle of the network. Linear coding [16] encodes the input data of each node using a linear combination, and decodes the matrix generated by linear combination of the various links of the network at the receiver node. Most practical solutions are based on Random Linear Network Coding (RLNC) [17], where a network node sends a random linear combination of stored packets.

Growth Codes proposed by Kamra et al. [3] improves data collection efficiency and provides good data protection in extreme environments through well-designed codeword conversion sequences and random distributed data replication. This paper designs two types of decoders, D-type decoder and S-type decoder. In addition, the dynamic degree distribution function is also designed. However, Growth Codes does not distinguish between data. Although the method of randomly exchanging data by nodes can improve the redundancy of data, it also reduces the efficiency of data collection. Reference [19] analyzes the factors affecting collection efficiency and the proportion of redundant symbols from a new perspective. A random feedback digest model RFDG [23] is proposed to digest redundant symbols, increase the effective information ratio in the network, and improve the
data decoding efficiency. Reference [22] pointed out that the performance of the growth code is not good in static scenarios. Various solutions have been proposed by studying the changes in the number and symbols of transmitted information and how to perform decoding. This solution can be used for static scenarios and improve the performance of data acquisition in dynamic scenarios. Literature [25] proposed local topology management of energy balance sub-loop.

D. APPLICATION OF INTELLIGENT ALGORITHM IN SENSOR NETWORK

Kim and Minkyu [20] introduced an evolutionary method for finding practical multicast protocols that provide all the advantages of network coding by reducing the number of coding nodes. The proposed method reduces the number of coding links relative to existing methods and is applicable to various general scenarios. Later, Kim and Minkyu [21] proposed a distributed GA algorithm, which makes the most time-consuming part of the calculation distributed in the network, and introduces greedy scanning to accelerate convergence. Literature [23] improves the convergence speed and improves the accuracy of node positioning by analyzing the error source of traditional algorithm in node localization and structural optimization and algorithm optimization for traditional BP network.

III. SYSTEM MODEL

A. PROBLEMS WITH TRADITIONAL DATA COLLECTION PROTOCOLS

Since the sensor network is used for disasters or emergencies, when certain data (such as fire point information) is generated, it needs to be quickly sent to the decision maker. Traditional data collection protocols such as Growth codes cannot distinguish between important data and ordinary data. This will result in important data and ordinary data being encoded and transmitted indefinitely by the node. Therefore, important data and ordinary data will be collected by the sink node at the same time. For important data, this collection efficiency is not tolerable.

In order to better understand the delays in the data collection protocol that cannot distinguish between important data and ordinary data, we use the Growth Codes Protocol (GCP) data collection protocol to “snap” the network data in each round to observe the amount of important data and ordinary data at the aggregation node.

Fig. 1 and Fig. 2 respectively show the relationship between the efficiency of the sink node receiving and decoding ordinary data and important data in the GCP and the required round, where the abscissa is the propagation round (this assumes that the sink node receives only one codeword in each round), and the ordinate is data recovery efficiency. As can be seen from the first and second figures, between 0 and 250 rounds, the sink node receives a data packet of 1 degree, and does not need to decode the received data, and the decoding efficiency of the data is high. Between 250 and 650 rounds, most of the data received by the aggregation node can be decoded, but fresh data is reduced. Between 650 and 800 rounds, since the sink node has decoded most of the source data, the likelihood that the sink node will receive the codeword and parse the new data is reduced, resulting in inefficient data recovery. For important data, in most cases you need to recover within 500 rounds. Under the GCP agreement, important data and ordinary data are all resolved in about 800 rounds. For important data with high real-time requirements, such delays are intolerable.

B. STARTING POINT

In order for important data to be collected quickly, traditional data collection protocols need to be improved. Growth Codes has good performance in data collection and provides effective protection for data in the network. This article is based on the Growth Codes, allowing the sink node to collect and decode important data faster.

This paper starts with the coding and exchange strategy, and realizes the rapid collection of important data on the basis of Growth Codes Protocol (GCP). Our goal is to improve the recovery efficiency of important data and to allow some of the ordinary data to be lost. In our model, nodes deal with important data and ordinary data differently when encoding and exchanging. For important data, we let it preempt the
coding opportunities of ordinary data. For different scenarios, we have designed two different schemes to allow important data to preempt the coding opportunities of ordinary data. When nodes exchange data, codewords with different important data content will also be treated differently. The protocol proposed in this paper can improve the collection efficiency of important data in the network and reduce the consumption of network resources.

C. IS FAIRNESS REALLY GOOD?

Traditional growth code-based data collection protocols assume that the data in the network is of equal importance, so the same strategy is used to process the codewords in the network. There is no problem in the case of equal importance of data in the network, but if two types of data are generated in the network and one type of data is more important than the other, these traditional protocols will not be sufficient.

Simply slowing down the delay can improve the recovery efficiency of important data, but it can’t solve the problem fundamentally. We believe that the key to solving this problem is to design a strategy to distinguish codewords and sacrifice some ordinary data encoding tunities. The component of important data is increased, so that the chances of decoding important data are also increased, and the codewords with high data content are specially processed in the forwarding process.

Therefore, we designed RTDCM. It divides the network into multiple layers, and the node allows important data to preempt the coding opportunities when encoding, and selects neighbors for data exchange according to the probability.

D. DATA PACKET STORAGE MODEL DESIGN

Since RTDCM distinguishes between important and ordinary data, we need to add a field to the packet so that the node recognizes which data is important. The protocol distinguishes packets with different important data contents, so it is necessary to bring the fields of important data to the packet. Fig. 3 visually depicts the packet model.

Compared with the traditional packet storage model, we have added two fields to the packet: Priority and Number, which represent the priority of the packet and the number of important data. When the priority is 0, it means that there is no important data in the data packet. When the priority is 1, it means that the data packet contains important data, and the number of important data is marked by the Number field.

In our design, the sensor knows whether the generated data is important data or ordinary data through state awareness to determine the priority field of the data packet. The node that generates important data encodes its important data into a data packet and updates the Number field of the data packet.

E. NETWORK CONFIGURATION

In order for the sensing node to send the codeword containing the important data to the sink node as soon as possible, we divide the entire network into layers. High value codewords can be exchanged to the sink node as soon as possible. The other configuration is the same as the GCP.

The network is divided into layers as shown in Fig. 4.

IV. REAL-TIME DATA COLLECTION MODEL DESIGN

A. SENSOR NODE PROBABILITY FORWARDING TABLE DESIGN

**Definition 4.1 Codeword value:** Given a codeword containing N original data symbols, where the number of key data symbols is a, the total amount of important data in the node is \( S = \sum_{i=1}^{a} a_i \). Then we call \( S = a_i \) the value of the codeword in the node cache.

**Definition 4.2 State Awareness:** The ability of a network sensor to know whether its perceived data is key or ordinary is called state awareness.

**Starting point:** We let the node have a weak trend in the choice of codewords, and transmit the codewords with higher codeword values to the sink node as soon as possible. Therefore, it is necessary to set a probability forwarding table according to different codeword values to select neighbors for data exchange.

The relationship between the forwarding probabilities of different value codewords and the decoding efficiency of important data is not clear. Therefore, we use the BP algorithm to fit the relationship and use the GA algorithm to find the optimal solution under the relationship. In order to adapt as quickly as possible, we set the forwarding table to 10 levels, corresponding to different codeword values, which is also a compromise between algorithm efficiency and may not be the best solution.
The form of the probability forwarding table is shown in Table 1, the probability of the codeword value is generated in non-descending order. The higher the codeword value, the higher the forwarding probability.

Algorithm 1 defines the generation process of the probabilistic forwarding table. First, traditional BP neural network is used to fit the relationship between the forwarding probability and the required rounds of all important data recovery. Then use the simple GA algorithm to find an optimal forwarding probability under the relational model.

**TABLE 1. Probability forwarding table.**

| Codeword value | Forwarding probability |
|----------------|------------------------|
| 0.1            | 0.021                  |
| 0.2            | 0.090                  |
| 0.3            | 0.158                  |
| 0.4            | 0.241                  |
| 0.5            | 0.340                  |
| 0.6            | 0.575                  |
| 0.7            | 0.780                  |
| 0.8            | 0.922                  |
| 0.9            | 0.976                  |
| 1.0            | 0.986                  |

**Algorithm 2 Neighbor Node Energy Estimation Algorithm**

1: The node perceives neighbors around itself and establishes a neighbor’s energy estimation table.
2: The initial energy consumption of each neighbor is $E_i=0$ (unit energy).
3: Exchange data with neighbors and update the energy estimation table:
4: If the data transmission is performed, the corresponding neighbor $E_i+E_i+0.03$ (unit energy).
5: if the data is received, the corresponding neighbor energy consumption value $E_i=E_i+0.01$ (unit energy).

**Algorithm 1 Probabilistic Forwarding Table Generation Algorithm**

**Fitting workflow:**
1: Initialize the connection weight and set the loss function $loss=\sum_{i=1}^{n}[y_i-o_i]$.
2: Set learning rate, maximum learning times.
3: Select the Kth input sample and its corresponding output: $x(k)=(x_1(k), x_2(k), \ldots, x_0(k))$.
4: Calculate the output, backpropagate according to the error, and correct the connection weight of each layer.
5: If the maximum number of learnings is reached, then output model.
6: else choose the next sample for training.

**Find the optimal forwarding probability workflow:**
1: Initialize the population, set the maximum round S, crossover rate c, and the mutation rate m.
2: Calculate fitness of each chromosome based on the model generated by the BP algorithm.
3: Use the betting wheel selection method to make a selection.
4: Perform crossover and mutation processes based on crossover and mutation rates.
5: If the maximum round is reached output the result.
6: else return to Step 2.

In order to extend the life of the entire network, we let the sensor node exchange data with nodes with less energy consumption in the neighbor nodes. Since the node needs to know the energy consumption of the neighbor node, it needs to exchange data once, which will waste the opportunity of code word exchange between nodes. Therefore, we design an energy estimation model, maintain an energy estimation table in the node, and update the energy consumption value of the neighbor in the neighbor energy estimation table when data is exchanged with the neighbor.

The neighbor node energy estimation algorithm is shown in Algorithm 2.

The form of the probability forwarding table is shown in Table 1, the probability of the codeword value is generated in non-descending order. The higher the codeword value, the higher the forwarding probability.

The neighbor node energy estimation algorithm is shown in Algorithm 2.

**Algorithm 2 Neighbor Node Energy Estimation Algorithm**

1: The node perceives neighbors around itself and establishes a neighbor’s energy estimation table.
2: The initial energy consumption of each neighbor is $E_i=0$ (unit energy).
3: Exchange data with neighbors and update the energy estimation table:
4: If the data transmission is performed, the corresponding neighbor $E_i=E_i+0.03$ (unit energy).
5: if the data is received, the corresponding neighbor energy consumption value $E_i=E_i+0.01$ (unit energy).

### B. SENSOR NODE ENERGY CONSUMPTION ESTIMATION TABLE DESIGN

In order to extend the life of the entire network, we let the sensor node exchange data with nodes with less energy consumption in the neighbor nodes. Since the node needs to know the energy consumption of the neighbor node, it needs to exchange data once, which will waste the opportunity of code word exchange between nodes. Therefore, we design an energy estimation model, maintain an energy estimation table in the node, and update the energy consumption value of the neighbor in the neighbor energy estimation table when data is exchanged with the neighbor.

**Algorithm 2 Neighbor Node Energy Estimation Algorithm**

1: The node perceives neighbors around itself and establishes a neighbor’s energy estimation table.
2: The initial energy consumption of each neighbor is $E_i=0$ (unit energy).
3: Exchange data with neighbors and update the energy estimation table:
4: If the data transmission is performed, the corresponding neighbor $E_i=E_i+0.03$ (unit energy).
5: if the data is received, the corresponding neighbor energy consumption value $E_i=E_i+0.01$ (unit energy).

**C. RTDCM BASIC FRAMEWORK**

**Definition 4.3 Level class:** The nodes are classified according to the level of the node, and the nodes in the same level are called the same level class nodes.

**Definition 4.4 Preempt loss:** A phenomenon in which important data preempts coding opportunities and causes a decline in the decoding rate of ordinary data.

We will design two RTDCM-based protocols. In order for the node to select a node with lower energy consumption when forwarding the codeword, we let the node store the energy consumption table of the neighbor node. The energy consumption table is estimated by the node and can be calculated without consuming communication. The RTDCM basic framework is shown in Algorithm 3.

**Algorithm 3 RTDCM Basic Framework Algorithm**

In the scenario of low preemption tolerance, we designed the RRTGCP (Random Real-Time Growth Code Protocol) based on probability preemption coding opportunities, and the other is PRTGCP (Priority Real-Time Growth Code Protocol) that preempts coding opportunities based on priority.

**D. RRTGCP**

In the scenario of low preemption tolerance, we designed the RRTGCP protocol. When the codeword is selected for encoding, the codeword with lower codeword value can also obtain the coding opportunity. According to the formula $Weight = c^p + K$, assign weights to the codewords in the buffer, where $p$ is the number of priority data in the current codeword, and $n$ is the total number of priority data in the buffer, $c$ and $K$ is the coefficient. And use the betting wheel to select the codeword that should be encoded in the current round. Based on this idea, We propose a new algorithm, as shown in Algorithm 4.
Algorithm 3 Framework of Real-Time Data Collection Model

Network and node settings:
1: Randomly broadcast \( N \) sensor nodes in the network.
2: Each node can store \( C \) codewords.
3: In each round, nodes only exchange data with neighbors.
4: In each round, the sink node will only receive a codeword from the neighbor.

Layered operation:
1: The sink node generates a hello packet containing the hierarchical information and broadcasts it to the neighbor.
2: if the node has received a hello packet, discard.
3: else write down the level information, update the hello packet, and broadcast.
4: If all nodes have determined their own level, then proceed to the next stage.

Sensor node workflow:
1: At initialization, the node senses data about the surrounding environment and generates data \( x_i \).
2: Use \( x_i \) to initialize the \( C \) storage space of the node.
3: Initialize maxdegree=1.
4: In the \( t \)-th round, the node \( i \) does the following work when selecting its own codeword:
5: Select the codeword \( x \) from the cache according to the preemption policy.
6: if degree(\( x \)) < MaxDegree, and \( x \) is not encoded with \( x_i \), then \( x = x \odot x_i \), else \( x=x \).
7: if the current round \( r > K_{\text{maxdegree}} \), maxdegree++.
8: In the \( t \)-th round, the node \( i \) does the following work when exchange codeword with neighbor:
9: Find the probability \( A \) corresponding to the codeword \( x \) in the probability table.
10: Probability \( A \) as the first-order probability of the low-level class.
11: Average the remaining probability \( 1 - A \) to other level classes.
12: Calculate the secondary probability based on the energy consumption of the nodes in each level class.
13: Integrate all secondary probabilities, use the gambling wheel to select neighbor nodes to exchange.
14: Exchange \( x \) with a neighbor's codeword \( y \).
15: Save \( y \) in where \( x \) was stored.

Sink node workflow:
1: The Sink node runs a D-type decoder and decodes the received codeword.

Algorithm 4 Random Real Time Growth Codes Protocol
1: Calculate the weight of each codeword based on the weight formula \( \text{Weight} = c \frac{p}{n+1} + K \).
2: Calculate the total weight according to the formula \( \text{sum} = \sum_{i=1}^{C} \text{Weight}(i) \).
3: Calculate the probability of a single codeword \( p_i \), and then calculate the cumulative probability \( c_i \).
4: Use the gambling wheel selection method to select codewords from the cache.
5: If degree(S)<MaxDegree, \( S=S \odot x_i \). If \( x_i \) is already included in the current code word \( S \), no encoding is performed.
6: If the current round \( k > K_{\text{MaxDegree}} \), then MaxDegree++.

Algorithm 5 : Priority Real Time Growth Codes Protocol
1: Calculate the number of important data in all codewords of the node \( S = \sum_{i=1}^{C} a_i \).
2: Calculate the value of the codeword according to the formula \( V = \frac{a_i}{C} \).
3: Sort according to the value of the codeword, and select the codeword with the highest codeword value.
4: If degree(S)<MaxDegree, \( S=S \odot x_i \). If \( x_i \) is already included in the current code word \( S \), no encoding is performed.
5: If the current round \( k > K_{\text{MaxDegree}} \), then MaxDegree++.

F. PROOF OF THEOREM
1) LEMMA1
Let the sink node have decoded \( r \) data symbols, the total number of data symbols is \( N \), and the sink node receives the codeword of \( d \) degree, then the probability that the sink can decode the new data symbol is:
\[
P = \binom{N-r}{d-r} \binom{r}{d} \frac{d}{N}
\]
Proof: If the received codeword of \( d \) degree can be solved by a new data symbol, then the codeword distance from the data symbol that has been solved is equal to 1. The data symbols of \( d-1 \) component codewords can be selected from \( r \) solved data symbols, the possible combinations are \( \binom{r}{d-1} \), and the remaining one of the new data symbols is from the remaining Among the undecided data symbols, the possible combination is \( \binom{N-r}{d} \), then according to the combination theory, the number of combinations of a \( d \)-degree decodable word is \( \binom{r}{d-1} \binom{N-r}{d} \), and the combined number of arbitrary codewords of \( d \) degrees is \( \binom{N}{d} \), so the probability that a new data symbol can be decoded for a \( d \)-degree codeword can be derived:
\[
P = \binom{r}{d-1} \binom{N-r}{d} \binom{N}{d}
\]
2) LEMMA2
Given a codeword sequence \( \delta_{[1,k]} = s_1, s_2, \ldots, s_k \), define \( |\delta_{[1,k]}| \) is the length of the sequence. It can then be concluded that the original data symbols decoded by the decoder \( D \)
based on the iterative decoding are not lower than the original data symbols decoded by the decoder S decoded in real time.

Proof: The S-type decoder first finds the codeword with the lowest degree from the sequence, and then tries to decode it. If it can’t decode, it discards. The probability that a codeword can be decoded is \( P = \left( \frac{1}{c} \right)^{N-1} \), if the number of symbols of the decodable number is \( \alpha \), if the decoding is possible, the mathematical expectation of the total number of codewords is: \( E(S) = \sum_{i=1}^{K} \alpha P \).

The D-type decoder does not discard codewords that cannot be decoded. In the case where one codeword can be decoded, the number of symbols that can be decoded is \( \beta \), and the mathematical expectation of solving the total number of codewords is: \( E(S) = \sum_{i=1}^{k} \beta P = \sum_{i=1}^{k} \frac{\alpha(N-1)}{N} \). Since \( \alpha = 1 \) and \( \beta \geq 1 \), \( E(D) \geq E(S) \), the number of original data symbols that the D-type decoder can decode is not lower than the S-type decoder.

3) LEMMA3

The total number of data symbols set in the overall network is \( N \), where the number of important data symbols is \( Q \), and the sink node has decoded \( r \) data symbols, wherein there are \( s \) important data symbols, and code words containing important data symbols. Preemptive coding opportunities increase the probability that important data will be decoded.

Proof: Preemptive coding opportunities result in an increase in the number of encoded packets containing priority data components throughout the network. Assuming that the current network contains \( m \) encoded data packets and the number of encoded data packets with priority data components is \( c \), the receiver can capture the content. The probability of encoding a packet of a priority data component is: \( P_1 = \left( \frac{1}{m} \right) = \frac{c}{m} \), the probability that sink can receive a codeword to decode a new data symbol is: \( P_2 = \left( \frac{1}{m} \right)^{N-1} \), the decoded new codeword new The probability that the data symbol is the priority data is: \( P_3 = \left( \frac{Q-s}{N} \right)^{c} \), then sink receives a codeword can solve The probability of a new priority data symbol is: \( P = \frac{c(Q-s)}{m(\frac{1}{m})} \), and as \( c \) increases, the probability \( P \) also increases. Therefore, the codeword preemption coding opportunity containing the priority data increases the probability that the priority data is decoded.

4) LEMMA4

Let the current sink node have decoded \( r \) data symbols, the total number of data symbols is \( N \), then the higher the content of important data in the codeword received by the Sink node, the higher the efficiency of decoding important data.

Proof: Assume that the current Sink receives a codeword of \( d \) degree, then by Lemma 1 the probability that it can decode the new data symbol is \( P = \left( \frac{d'^{(N-1)}}{Q} \right) \), assuming that the ratio of important data symbols is \( \alpha \) in the current d-codeword, the probability that the current codeword decodes important data is: \( P_\alpha = \left( \frac{d'^{(N-1)}}{Q} \right) \). Assuming that the Sink node continuously collects \( K \) rounds and receives one codeword per round. The average number of codewords solved per round is \( c \), then the expected number of important data decoded in the \( K \) round is: \( E(i) = \sum_{i=1}^{K} cP_\alpha = \sum_{i=1}^{K} \frac{\alpha(N-1)}{N} \). \( E(i) \) also increases as \( \alpha \) increases, which also indicates that the higher the content of important data in the codeword received by the Sink node, the higher the efficiency of decoding important data.

V. PERFORMANCE ANALYSIS

In this section we will compare the performance of two RTDCM-based protocols and evaluate them in a disaster environment.

A. EXPERIMENTAL PARAMETER SETTING

In the experiment, we randomly placed 500 sensor nodes in the \( 100 \times 100 \) surveillance space, of which 50 nodes were used to generate important data and 450 nodes were used to generate ordinary data. The nodes have the same communication radius \( R \) and have a storage capacity of 10 code words.

To better evaluate our proposed agreement, we will start with the following performance indicators:

(1) Recovery Rate of Important Data: It can best reflect the efficiency of the protocol to collect and recover important data. If the protocol uses a relatively small number of rounds to recover all important data, it indicates that the protocol is highly efficient.

(2) Proportion of ordinary data recovery: The protocol we propose not only focuses on the recovery of important data, but also on the recovery of ordinary data. Under the premise of all important data recovery, the more ordinary data recovery, the better the protocol performance.

(3) Overall Network Energy Consumption: Network energy consumption is a problem that all sensor network-based protocols need to consider. For our agreement, we not only need to assess the efficiency of their data recovery, but also to assess their energy efficiency. Since the data transmission of the node occupies most of the energy consumption of the node, other energy consumption is ignored in the energy consumption evaluation, and only the energy consumption of the data transmission part is considered.

(4) Data Protection in the Event of a Disaster: Sensor networks are usually placed in extreme environments and nodes can be damaged at any time. In the event of node damage, the data protection of the protocol is critical.

B. DATA RECOVERY EFFICIENCY IN STABLE SCENARIOS

In this section we focus on the recovery efficiency of important data and ordinary data, the main comparison objects are GCP, RRTGCP, PRTGCP.
1) IMPORTANT DATA RECOVERY EFFICIENCY OF EACH PROTOCOL

Fig. 5 and Fig. 6 mainly study the key data recovery efficiency of the three protocols in the case of communication radius of 10, 30 (sparse network, dense network). View the important data collection efficiency of different protocols under different communication radius conditions.

In dense network environments and sparse network environments, endpoint data collection efficiency of RRTGCP and PRTGCP is better than GCP. The reason is that RRTGCP and PRTGCP perform differentiated processing on critical data and non-critical data, improve the recovery efficiency of important data, and achieve satisfactory recovery efficiency in sparse networks.

2) COMPARISON OF NODE COMMUNICATION RADIUS AND IMPORTANT DATA COLLECTION EFFICIENCY

As shown in Fig. 7, as the communication radius increases, the number of rounds required for all three important data collections is reduced. The main reason is that as the node communication radius increases, the number of neighbors of data that can be exchanged around the node also increases, the connectivity between the nodes increases, and the node data can be transmitted to the receiving node at a faster rate. It can also be seen from the figure that GCP is greatly affected by the connectivity of the nodes, and RRTGCP and PRTGCP are less affected by the connection of the nodes. As the node communication radius increases, the GCP important data collection efficiency rises faster. Therefore, compared with RRTGCP and PRTGCP, GCP has low tolerance to node connections.

3) COMPARISON OF ORDINARY DATA RECOVERY EFFICIENCY UNDER EACH PROTOCOL

As shown in Fig. 8, the GCP protocol cannot distinguish between important data and ordinary data. When the important data is received, the ordinary data is also received, and the ordinary data does not change with the node communication radius. RRTGCP and PRTGCP allow codewords containing important data to preempt codewords of other codewords, thereby increasing the proportion of coded packets containing important data in the overall packet. When exchanging data, it is likely to exchange data with low-level neighbors. At the Sink node, codewords containing more important data can be collected. The proportion of important data in the channel of the Sink node increases, and the proportion of common data decreases, resulting in low efficiency of collecting ordinary data.

When the node communication radius is too small, the Sink node needs more rounds to collect all the important data.
4) COMPARISON OF ENERGY CONSUMPTION OF EACH PROTOCOL

As shown in Fig. 9, as the communication radius increases, the total network energy consumption of the three protocols decreases. The energy consumption of the three protocols in the node communication radius between 20 and 30 tends to be flat. And compared to the traditional GCP, the other two protocols have a significant reduction in energy consumption.

C. DATA RECOVERY EFFICIENCY IN DISASTER SCENARIOS

The transmission scheme designed in this paper can be used to detect accidents such as fire and mechanical damage in industrial production. If some unexpected conditions are detected, this part of the information can be transmitted in time, and other data can be collected. In an industrial environment, sensor damage is unpredictable. This section uses simulation to simulate a disaster scenario to verify the performance of this transmission scenario in a disaster scenario.

1) COMPARISON OF KEY DATA RECOVERY EFFICIENCY IN CENTRALIZED DISASTER SCENARIOS

As shown in Fig. 10, RRTGCP and PRTGCP performed well in both early and late disaster scenarios in a centralized disaster scenario. The main reason is that RRTGCP and PRTGCP allow key data preemption coding opportunities, select codewords from the node’s cache through probability selection and priority selection, and the node passes the encoded data packets to lower-level nodes with greater probability according to the policy. This allows codewords containing important data to be passed to other nodes (possibly low-level nodes or high-level nodes) before a disaster occurs, greatly increasing the survival rate of important data.
2) COMPARISON OF KEY DATA RECOVERY EFFICIENCY UNDER RANDOM DISASTER SCENARIOS

As shown in Fig. 11, RRTGCP and PRTGCP performed well in both early and late disaster scenarios in a random disaster scenario. The main reason is that RRTGCP and PRTGCP allow key data to preempt coding opportunities and exchange data from other nodes, thus enhancing the redundancy of key code words. However, this also deprives the ordinary codewords of the opportunity to pass to other nodes to a certain extent, resulting in the decline of the protection of ordinary codewords.

3) THE RECOVERY EFFICIENCY OF KEY DATA VARIES WITH THE RADIUS OF THE DISASTER

It can be seen from Fig. 12 that the disaster radius has a greater impact on the rounds required for GCP data recovery, but less on RRTGCP and PRTGCP. The main reason is that RRTGCP and PRTGCP prioritize the encoding of important data and the exchange with neighbors, which makes important data more redundant and more resistant to disasters.

4) ORDINARY DATA RECOVERY EFFICIENCY VARIES WITH THE RADIUS OF THE DISASTER

As can be seen from Fig. 13, in the case of disasters, the recovery efficiency of ordinary data in RRTGCP and PRTGCP is lower than that of GCP, mainly because important data captures the coding opportunities of ordinary data, resulting in a decrease in the protection of ordinary data. Moreover, since PRTGCP uses a priority preemption strategy, it further leads to a decline in the protection of ordinary data. When the disaster radius is 50, because RRTGCP adopts a layering strategy, the data propagation has a certain directionality, and the data can be transmitted to the aggregation node as soon as possible, which is also beneficial to the protection of the data.

VI. CONCLUSION AND FUTURE WORKS

In this article, we first analyze the problem that traditional data collection protocols deal with different types of data fairly and cause important data collection to be delayed. We proposed RTDCM from the perspective of network layering and preempting coding opportunities, and proved the feasibility of the model through data derivation and experiments. The model first divides the network into multiple layers so that the sensing nodes know the location of the convergent nodes. By allowing codewords with high codeword values to preempt coding opportunities for other codewords, the proportion of codewords containing important data in all codewords is increased. Neural network algorithm and genetic algorithm calculate the node data exchange probability of codewords corresponding to different codeword values to avoid random forwarding during data exchange. At the same time, considering the energy consumption of the entire network, a neighbor node energy consumption estimation table was designed to allow nodes to consider the energy consumption of the nodes during data exchange, thereby effectively reducing the energy consumption of the entire network. Based on this model, two data collection protocols with different preemption strategies are designed. Experiments show that these two protocols based on RTDCM can improve the decoding efficiency of important data and reduce the delay of important data collection.

Two RTDCM-based data collection protocols can improve the collection efficiency of important data, but reduce the collection efficiency of ordinary data to a certain extent. It also needs some other operations to divide the network into multiple layers, which will increase the burden on the network. This protocol is only applicable to networks with data of two different importances. It does not apply to networks with multiple priority data. In future work, we hope to further improve the preemption strategy to restore ordinary data as much as possible, and improve data exchange rules to introduce multi-priority data collection protocols.

REFERENCES
[1] Y. Nishikawa, T. Sasamura, Y. Ishizuka, S. Sugimoto, S. Iwasaki, H. Wang, T. Fujishima, T. Fujimoto, K. Yamashita, T. Suzuki, and K. Kurihara, “Design of stable wireless sensor network for slope monitoring,” in Proc. IEEE Topical Conf. Wireless Sensors Sensor Netw. (WiSNet), Anaheim, CA, USA, Jan. 2018, pp. 8–11.
[2] Y. Zhang, L. Sun, H. Song, and X. Cao, “Ubiquitous WSN for healthcare: Recent advances and future prospects,” IEEE Internet Things J., vol. 1, no. 4, pp. 311–318, Aug. 2014.
[3] A. Kamra, V. Misra, J. Feldman, and D. Rubenstein, “Growth codes: Maximizing sensor network data persistence,” in Proc. ACM SIGCOMM Comput. Commun. Rev., vol. 36, 2006, pp. 255–266.

[4] C.-Y. Wan, S. B. Eisenman, A. T. Campbell, and J. Crowcroft, “Siphon: Overload traffic management using multi-radio virtual sinks in sensor networks,” in Proc. 3rd Int. Conf. Embedded Netw. Sensor Syst., 2005, pp. 116–129.

[5] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Medard, and J. Crowcroft, “XORs in the air: Practical wireless network coding,” IEEE/ACM Trans. Netw., vol. 16, no. 3, pp. 497–510, Jun. 2008.

[6] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, “Network information flow," IEEE Trans. Inf. Theory, vol. 46, no. 4, pp. 1204–1216, Jul. 2000.

[7] J. W. Byers, M. Luby, M. Mitzenmacher, and A. Rege, “A digital fountain approach to reliable distribution of bulk data,” in Proc. ACM SIGCOMM Conf. Appl., Technol., Archit., Protocols Comput. Commun., New York, NY, USA, 1998, pp. 56–67.

[8] M. Luby, “LT codes,” in Proc. 43rd Annu. IEEE Symp. Found. Comput. Sci., Jun. 2002, pp. 271–280.

[9] A. Shokrollahi, “Raptor codes,” IEEE/ACM Trans. Netw., vol. 14, no. 6, pp. 2551–2567, Jun. 2006.

[10] J. Mcdonagh, M. Sala, A. O’Allmhrarian, V. Katewa, and E. Popovici, “Efficient construction and implementation of short LDPC codes for wireless sensor networks,” in Proc. 18th Eur. Conf. Circuit Theory Design, Aug. 2007, pp. 703–706.

[11] Y. Lin, B. Liang, and B. Li, “Data persistence in large-scale sensor networks with decentralized fountain codes,” in Proc. IEEE INFOCOM 26th IEEE Int. Conf. Comput. Commun., 2007, pp. 1658–1666.

[12] A. Taliari and N. Rahnavard, “Robust LT codes with alternating feedback,” Comput. Commun., vol. 49, pp. 60–68, Aug. 2014.

[13] H. D. T. Nguyen, L.-N. Tran, and E.-K. Hong, “On transmission efficiency for wireless broadcast using network coding and fountain codes,” IEEE Commun. Lett., vol. 15, no. 5, pp. 569–571, May 2011.

[14] X.-Y. Liu, Y. Zhu, L. Kong, C. Liu, Y. Gu, A. V. Vasilakos, and M.-Y. Wu, “CDC: Compressive data collection for wireless sensor networks,” IEEE Trans. Parallel Distrib. Syst., vol. 26, no. 8, pp. 2188–2197, Aug. 2015.

[15] E. Candes, J. Romberg, and T. Tao, “Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information,” IEEE Trans. Inf. Theory, vol. 52, no. 2, pp. 489–509, Feb. 2006.

[16] S.-Y. R. Li, R. W. Yeung, and N. Cai, “Linear network coding,” IEEE Trans. Inf. Theory, vol. 49, no. 2, pp. 371–381, Feb. 2003.

[17] T. Ho, M. Medard, R. Koetter, D. Karger, M. Effros, J. Shi, and B. Leong, “A random linear network coding approach to multicast,” IEEE Trans. Inf. Theory, vol. 52, no. 10, pp. 4413–4430, Oct. 2006.

[18] P. Ostovari and J. Wu, “Fault-tolerant and secure distributed data storage using random linear network coding,” in Proc. 14th Int. Symp. Modeling Optim. Mobile, Ad Hoc, Wireless Netw. (WiOpt), 2016, pp. 1–8.

[19] W. Zhang, J. Chang, F. Xiao, Y. Hu, and N. N. Xiong, “Design and analysis of a persistent, efficient, and self-contained WSN data collection system,” IEEE Access, vol. 7, pp. 1068–1083, 2019.

[20] M. Kim, C. W. Ahn, M. Medard, and M. Effros, “On minimizing network coding resources: An evolutionary approach,” in Proc. NetCod, 2006.

[21] M. Kim, M. Medard, V. Aggarwal, U.-M. O’Reilly, W. Kim, C. W. Ahn, and M. Effros, “Evolutionary approaches to minimizing network coding resources,” 2007, arXiv:cs/0702037 [Online]. Available: https://arxiv.org/abs/cs/0702037

[22] D. Munaretto, J. Widmer, M. Rossi, and M. Zorzi, “Network coding strategies for data persistence in static and mobile sensor networks,” in Proc. 5th Int. Symp. Modeling Optim. Mobile, Ad Hoc Wireless Netw. Workshops, 2007, pp. 1–8.

[23] Y. Yu and L.-H. Zhang, “WSN location method based on BP neural network in NLOS environment,” in Proc. Int. Conf. Wireless Commun. Sensor Netw., 2014, pp. 321–325.

[24] M. Magni, D. Boyle, D. Brunelli, E. Popovic, and L. Benini, “Ensuring survivability of resource-intensive sensor networks through ultra-low power overlays,” IEEE Trans. Ind. Informat., vol. 10, no. 2, pp. 946–956, May 2014.

[25] W. Zhang, N. Xiong, L. T. Yang, G. Jia, and J. Zhang, “BCHED-energy balanced sub-round local topology management for wireless sensor network,” J. Internet Technol., vol. 13, no. 3, pp. 385–394, 2012.

[26] M. Satyanarayanan, “Fundamental challenges in mobile computing,” in Proc. 15th Annu. ACM Symp. Principles Distrib. Comput., May 1996, pp. 1–7.