

Research Article

Fishery Data Distribution System Based on Distance Prior Network Coding Strategy with Buffer Mapping Mechanism

Wei Zhang , Xiarui Li , Renjie Zhou , and Yangyang Bai 

Department of Computer Science and Technology, Hangzhou Dianzi University, Hangzhou 310018, China

Correspondence should be addressed to Xiarui Li; lixiarui@hdu.edu.cn

Received 14 January 2021; Revised 10 April 2021; Accepted 20 April 2021; Published 7 May 2021

Academic Editor: Paolo Bellavista

Copyright © 2021 Wei Zhang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Marine fishery production safety has always been a livelihood issue of high concern to government departments at all levels in China. With the construction of marine informatization, a large number of fishing vessels have entered the Vessel Monitoring System (VMS). The massive, highly concurrent, and continuous track data generated by these vessels has posed great challenges to real-time data distribution. To cope with low distribution efficiency in the original data distribution system, this paper proposed a data distribution model based on the combination of network coding and UDP, which improves both reliability and efficiency of data distribution. In order to further enhance the efficiency of data distribution, a Codeword Distance Priority Protocol based on Buffer Map (CDPPBM) was added to the proposed model for the concentration of innovative codewords in the network, thereby increasing the effectiveness of data received by nodes. Experimental results show that the protocol proposed in this paper improves the data distribution efficiency by about 75% on average, compared with the LT code. The influence of data block size on network coded data distribution system is not involved in the previous work. Therefore, this paper discusses the Block to Piece Protocol (BPP) for large files, divides large files into fixed sizes equally, and distributes n data blocks multiple times to find the optimal piece size. The experimental results show that there is an optimal n for large files of different sizes, which can maximize the efficiency of data distribution.

1. Introduction

The modernization and informatization of marine fisheries, which play an important role in the marine industry, have attracted the intense attention of many coastal countries in recent years. The fishing vessel supervision system, as one of the most important means of fisheries informatization, has been utilized widely in the fields of fishing vessel navigation, safety rescue, fishery production, and marine monitoring [1]. Beidou VMS can collect data in real time and transmit it through satellite communication technology to the ground receiving station, which needs to send these data to various users in real time. As the number of users and the amount of data are increasing, the load on the ground receiving station increases, which may overwhelm the equipment and resources.

In this paper, a descriptive analysis was performed on the push failure of a certain enterprise fishing boat positioning push server. At present, the fishing boat positioning push server consisted of a source node and multiple receiving nodes, in which the source node was responsible for receiving, analyzing, and storing the serial data returned by Beidou and then forwarding them to multiple receiving nodes. The original system adopted the C/S data distribution model based on the TCP communication protocol. Assuming that the source node sends a file composed of m blocks of data to N clients, the source node needs to send at least $m * N$ rounds so that all clients can receive the complete file. The data distribution mode of the original system has the following two shortcomings: the excessive pressure on the source node and the handshake characteristic of TCP.

In response to the above problems, this paper does the following work:

- (i) In the original system (OS), the source node bears most of the load pressure, and the client is almost idle. In order to solve this problem, inspired by the literature [2], this paper introduces a coding strategy in the P2P network. There would always be some rare data that some nodes cannot access in the later stage of data distribution. In order to solve this problem, the source node of the system continuously distributed data to receiving nodes using fountain codes based on direct transmission combined with robust soliton degree distribution [3]. The Codewords Degree Control Protocol (CDCP) was employed for data encoding exchange between receiving nodes.
- (ii) In order to increase the probability of obtaining a valid codeword in each data exchange, this paper proposed a strategy to exchange Buffer Map information with neighboring nodes at a certain frequency. The node searches the Buffer Map of the neighboring node and selects the appropriate data packet to send it to the neighboring node according to the principle of codeword distance priority, so that the data packet received by the neighboring node is highly likely to be valid, thus improving the efficiency of data distribution.
- (iii) This paper also considers the effect of piece size on system data distribution efficiency when the size of the file to be sent is determined. To this end, this paper discusses the Block to Piece Protocol (BPP) for large files, divides large files into fixed sizes equally, and distributes n data blocks multiple times to find the optimal piece size.

The rest of this paper is structured as follows. In Section 2, the data distribution related works are introduced briefly. The network model and some related configurations are presented in Section 3. Moreover, the specific scheme is described in Section 4. In Section 5, the experimental results are analyzed. Finally, some conclusions and prospects are drawn in Section 6.

2. Related Work

With widespread Internet access, applications such as file downloads and video-on-demand have led to exponential growth in Internet. Since then, the traditional model has overwhelmed the server. The application of content distribution network (CDN) [4] reduces the data transmission pressure of the backbone network, but the edge network is still a model. The P2P network makes use of the processing power of the client to greatly reduce the dependence on the server and change the current server-centered state of the Internet. However, obtaining all data blocks from the client is like a process of philately where there are always some scarce data blocks, making it a difficult process. In the study of improving network data transmission performance,

network coding has important theoretical value and broad application prospects. In terms of the data collection protocol, the coding technology represented by the digital fountain code [5] is widely used in data collection and storage in wireless sensor networks. The decoding algorithm of LT code [3], which is a rateless linear random fountain code, is simple and efficient but is carried out at a fixed cost. Kamra et al. [6] first conducted a systematic analysis of the data persistence of data collection without configuration disaster scenarios and proposed incremental codes, Growth Codes. The main idea is to gradually integrate the received data with the data stored by itself, increase the “degree” of data, and then exchange these data codewords with neighboring nodes, but there is a problem of redundant data repeated transmission. Reference [7] analyzes the factors affecting the collection efficiency from a new perspective, the ratio of redundant symbols. A random feedback digest (RFDG) model is proposed to digest redundant symbols, increase the effective information ratio in the network, and improve the efficiency of data decoding. In addition, the application of network coding in wireless communication, including wireless sensor networks, wireless ad hoc networks, and wireless mesh networks, is also a research hotspot. In wireless ad hoc networks, network coding can be used to increase network throughput, reduce node energy consumption, and extend the entire network life cycle [8–11]. Besides, in wireless sensor networks, network coding is used for effective data gathering [12–14]. The application-level multicast, information security, distributed storage, data processing [15], network layering, and peer influence [16] have also been studied accordingly.

The application of network coding in P2P network has also attracted many scholars’ attention. In Magnetto’s work [2], rateless codes are used as the key content delivery mechanism for the design of a novel P2P live streaming application. Gkantsidis and Rodriguez first proposed the application of network coding to P2P content distribution systems. They designed Avalanche, a file distribution system based on network coding [17] which partitions the original file into blocks and employs a random network coding algorithm to encode and distribute them to the network P2P. The nodes in the network also encode the received data blocks, then forward them until enough linearly independent data blocks are received, and restore the original data file by decoding. Ma [18] et al. claimed that the performance of the system using sparse network coding technology was slightly improved compared with that without network coding. Reference [19] developed an effective packet selection mechanism, called Intelligent Packet Coding (IPC), which further improves the efficiency of content distribution based on network codes in P2P networks. In addition, Xu et al. [20] studied the relationship between the scheduling load and the coding load of the content distribution system based on network coding and proposed a P2P content distribution that combines the combination of “local rarest first” scheduling (network rarest first) and network coding algorithm. Wan et al. [21] introduced the particle swarm optimization algorithm in order to solve the optimal scheduling problem of grouping. In order to solve problems

such as waiting time, methods based on crowdsourced measurement have been extensively studied [22]. However, Wang and Li [23] questioned the usefulness of network coding in content distribution. Similarly, Chiu [24] and others claimed that there is no coding gain in network coding in content distribution networks.

Some previous work has investigated the impact of data distribution fragment size on other peer-to-peer content distribution systems. In version 3.1 of the official BitTorrent implementation, although no specific reason was given, the default fragment size was reduced from 1 MB to 256 KB. Presumably, the performance advantages of smaller fragments were noticed. Hoßfeld et al. [25] used simulation to evaluate different fragment sizes in eDonkey-based mobile file sharing systems. They found that the download time decreased as the fragment size increased. The authors of Dandelion [26] evaluated the performance of fragments of different sizes and mentioned that the TCP effect is a potential cause of poor performance of small fragments. The authors of Slurpie [27] briefly mentioned the tradeoff in fragment size and mentioned that TCP overhead is a disadvantage of small fragments. Marciniak et al. [28] introduced the results of actual experiments with different fragment sizes on a controlled BitTorrent test bed, proving that fragment size is critical because it determines the degree of parallelism in the system. These works have not yet involved the effect of fragment size on distribution efficiency in a data distribution system based on network coding.

Most of the existing methods proposed here cannot control the transmission of redundant traffic in the transmission based on network coding, which will increase the network load and reduce the network throughput. This paper attempts to alleviate the above problems. What this article needs is a weak real-time data distribution system. Each receiving department needs to monitor the fishing situation data in real time, and the source node needs to send data to each receiving node without interruption. Therefore, this article only needs to consider how to make the total time for the target file to reach all receiving nodes the shortest, and the decoding completion time of each receiving node is similar; that is, “the number of rounds required by the network for all peers to obtain all the information” will be used as evaluation indicators.

3. System Model

In this section, in order to solve the above problems, we will establish a network model whose application scenario is that the source node uses an efficient data distribution method to make the fixed receiving node obtain the complete file as soon as possible. Table 1 lists the notations used in this paper.

3.1. Overview of System Model. This paper builds the network into a directed graph $G = (N + 1, L)$, where $N + 1$ represents a source node and N receiving nodes in the network, and L represents a UDP communication link between terminals.

It is assumed that any node in the network can establish an overlay connection with any other nodes and the overall

model has a three-layer structure. The source node S in the network is responsible for receiving and processing the data from the satellite and then distributing it to the nodes in the form of sequence blocks. N forwarding nodes R_1, R_2, \dots, R_N are responsible for encoding and forwarding. When these encoded blocks are transmitted between any two forwarding nodes, the node either downloads the blocks completely or does not at all. N receiving nodes D_1, D_2, \dots, D_n correspond to N forwarding nodes. The relationship between the source node and the forwarding node is 1: N , the relationship between the forwarding node and the receiving node is 1: 1, and the forwarding nodes can communicate with each other. In order to improve the efficiency of network data transmission, the system abandoning the TCP transmission protocol adopts the UDP transmission protocol and introduces network coding to ensure the reliability of network data transmission. At present, in order to improve the reliability of data transmission, forward error correction (FEC) technology is generally used. The so-called forward error correction technology is used to perform forward error correction coding on the source data before sending data packets, and certain redundant data packets are used to repair packet loss and ensure the reliability of data transmission. However, the forward error correction code introduces redundant data packets and increases the cost of decoding, which may cause unnecessary waste, and the biggest disadvantage of traditional FEC is that the error correction ability is not strong. There are many research directions for forward error correction codes. Among them, the representative LT code in the digital fountain code is used in this paper. Its main characteristics are strong error correction ability and being rateless, which can generate any number or even an infinite number. After the source node encodes the data, the receiving node can complete the decoding after receiving a certain number of encoded data packets and restore the original data. Therefore, the LT coding strategy can solve the unreliability of UDP well.

Figure 1 shows the network structure model of the system. The source node divides the received file F into m equal-sized blocks $F_i (i = 1, 2, \dots, m)$ which are called metadata pieces. Each piece of metadata pieces is a piece of fishing information. The data used in this experiment is the position information of fishing boats, which is mainly text information composed of time, longitude, and latitude, and each piece of positioning information is about 68 bits. The source node S sends a data packet encoded by $F_i (i = 1, 2, \dots, m)$ to the forwarding node $R_i (i = 1, 2, \dots, N)$. After receiving the data, the node R_i starts data exchange with neighboring nodes. When all forwarding layer nodes receive the complete file F , the source node ends the distribution of file F and begins the transmission of the next file. By default, the system forwarding layer node receives a complete file, which means that the receiving terminal can receive the complete file. Therefore, the subsequent discussion and research work is only carried out between the source node and the forwarding layer node.

How to measure the efficiency of system distribution? It is advisable to measure the time according to the time slot proposed in the literature [29–33]. The time of a node

TABLE 1: Explanation of notations.

Notations	Explanation
m	The amount of metadata
S	The source node
R_i	The forwarding node
D_i	The receiving node
F_i	The i th metadata
M	The round sent by the source node
N	The number of nodes
W_1	The expected number of data packets that the node can actually receive
W_2	The expectation of the number of data packets that the node needs to receive
ϵ	The additional codewords that the node needs to receive
d	The codeword degree
r	The amount of decoded metadata
Re_i	The mathematical expectation to recover the number of codewords required for Re_i metadata is K_i
K_i	The mathematical expectation of the number of codewords required for degree i

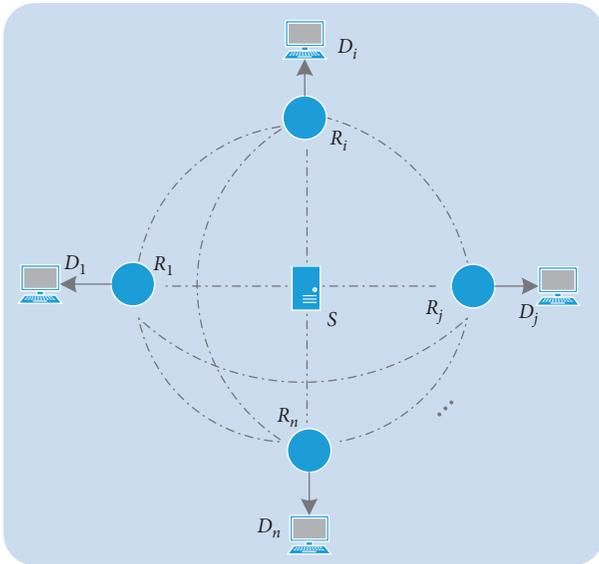


FIGURE 1: Network structure model.

(including the source node and forwarding node) to transmit a data block is regarded as a time unit. Therefore, the number of data blocks uploaded by the source node can be used as an indicator to measure the system, namely, the number of sending rounds of the source node.

3.2. Problems with Traditional Data Transmission Protocols.

This system is a weak real-time system, and the satellite will continuously transmit data to the source node. Therefore, the source node needs to continuously transmit data according to the upper-layer application file. In order to improve the efficiency of data transmission, the source node needs to continuously send encoded packets to the node. When the node decodes the complete file F , it will send a feedback message to the source node. After the source node receives feedback from all nodes, it will stop sending files. This process is similar to the that of fountain codes.

The robust soliton distribution (RSD) coding strategy proposed by LT [3] continuously changes the degree of

codewords in the overall network, so that a codeword will contain multiple-source data information, thereby improving data transmission efficiency. However, when LT encodes and transmits data according to the robust soliton distribution coding strategy, the decoding curve will show relatively obvious “cliff effect” and RSD will generate a large number of 2° codewords. At the beginning of transmission, due to the lack of 1° codewords, the codewords received by each node cannot be decoded immediately. After the node accumulates a certain number of codewords and receives a certain number of 1° codewords, the node’s decoding rate rises sharply.

In order to better understand the problem of low node decoding rate in the early stage caused by the lack of 1° packet in RSD system, experiments were carried out on RSD. The source file is concerning specific fishing information, and a piece of information is a piece of metadata pieces. The number of nodes is $N = 3$, and the file size is $m = 100$ metadata pieces. After many experiments, the average value is obtained. The relationship between the decoding situation and the number of rounds is shown in Figure 2.

The X -axis shows the number of sending rounds of the source node, and the Y -axis indicates the number of data pieces decoded by the node in the current round. It can be seen from this that, between rounds 0 and 120, the node obtains fewer 1° codewords and may receive a large number of high codewords. Therefore, the decoding operation cannot be performed, and the decoding efficiency of the data is relatively low. Between rounds 120 and 250, the node received a certain degree of codeword and accumulated a certain amount of data. Therefore, most of the received data can be decoded, and this stage has a higher decoding efficiency. Between rounds 250 and 320, because the node has decoded most of the source data, the probability of the node receiving a codeword and solving fresh data decreases, resulting in a low data decoding rate at this stage.

3.3. Degree Distribution Design of Source Node. From the above analysis, it can be seen that the decoding delay problem of nodes in the initial stage under the RSD protocol is more obvious. The important reason is that the degree

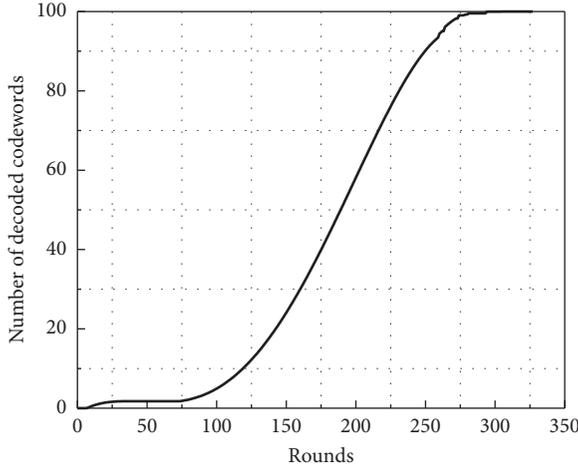


FIGURE 2: The amount of decoded data of a node under the RSD protocol changes with the sending round of the source node.

distribution in the RSD protocol is relatively high, and there are not enough 1° codewords in the initial stage to trigger decoding algorithm. Therefore, in order to reduce the “cliff effect,” this paper proposes a direct transmission combined with a robust soliton distribution scheme. On the basis of ensuring good data coverage, it reduces the average degree of codewords in the system, reduces coding overhead, and increases decoding rate, thereby increasing data transmission efficiency. The source node S first sends metadata pieces F_1, F_2, \dots, F_m to each node in turn (F_i is sent in the i th round) to increase the concentration of packets with a degree of 1 in the network. In the $m + 1$ th round, the RSD is adopted, and the degree of the encoded packet is selected according to the probability. Refer to Algorithm 1 for specific design.

3.4. Forwarding Layer Design

3.4.1. Initial Design of Forwarding Layer. Before the current generation of file distribution, all forwarding layer nodes are empty. The source node can only communicate with one node in a round. If the node of the forwarding layer randomly selects neighbor node for data exchange, there will be a certain probability that two nodes with no data will always make requests to each other, resulting in a large number of invalid requests and wasting communication overhead. If all nodes get at least one piece of data, the source node needs to send rounds in two extreme cases:

- (i) *The Worst Case.* The first piece of data of all forwarding layer nodes comes from the source, so it needs to go through N rounds before the last node can get the first data.
- (ii) *The Most Ideal Situation.* As shown in the following figure, if a certain criterion is followed, the source can let all nodes get a piece of data with the least number of transmission rounds.

Before all nodes have a piece of data, the data packet sent by the source node carries the address of the neighbor node

that the receiving node needs to communicate with in the next few rounds. According to this rule, all nodes can obtain a piece of data in the shortest round, reducing the invalid communication overhead.

According to the above forwarding rules, there are $L_1 = 1$ node with data in the first round and $L_2 = (1 + 1) + 1 = 2L_1 + 1$ nodes with data in the second round. In the v th round, $L_v = 2L_{v-1} + 1$ nodes have data. When $L_{v-1} < N \leq L_v$, the system needs to go through at least $v_{\min} = v$ rounds to make each node have a piece of data.

3.4.2. Design of Data Exchange Protocol Based on Codeword Degree Control

(1) *Design of Data Coding and Exchange at Forwarding Layer.* The experimental proof in the design of source node degree distribution shows that if the data is exchanged between nodes only through simple replicating and forwarding, it will lead the nodes to wait for the last one or two pieces of data in nearly half of the rounds, causing serious “coupon” collection problem. The reason may be that the codeword received by the node is probably a low-degree codeword (1 or 2°), the probability of low-degree inclusion of innovative codewords will be much lower than that of high-degree packets; as a result, nodes will need to receive a large number of codewords to get the last few data pieces.

On account of the above analysis, this paper designed a data coding exchange protocol based on codeword degree control. In the light of the idea of degree of time conversion sequence in Growth Codes proposed by Kamra et al. [6], the randomness is reduced by controlling the degree of flowing packets between nodes according to the rounds, thus reducing the flow of invalid packets and improving the efficiency of data distribution.

We will start from the node’s actual expectations of receiving data packets and the number of data packets that need to be received, and deduce the relationship between the round of the source node’s need to send packets and the codeword degree control sequence.

The expected number of data packets that the node can actually receive is W_1 . In our forwarding strategy, when node A sends a request to neighbor node B , it will add an encoded data packet to the request and send it as a “gift” to neighbor node B . After receiving the request, the neighbor node B will respond to the request and send a coded packet back to the node A . Therefore, each node can receive an average of 2 data packets sent from neighbor nodes in each round.

Assuming that after the source sends M rounds, all nodes decode the complete file F , then the average number of packets received by each node is W_1 :

$$W_1 = 1 + 2(M - v_{\min}) + \frac{M - v_{\min}}{N}, \tag{1}$$

where 1 means that when the forwarding layer is initialized, each node will have a piece of data obtained from the source node or other neighbor nodes; $2(M - v_{\min})$ means, after

- (1) First, the source node S sends metadata F_1, \dots, F_m to each node in turn (F_i is sent in round i)
- (2) Calculate the ideal soliton degree distribution $\omega(d) = \begin{cases} (1/m) & d = 1 \\ (1/d(d-1)) & d = 2, 3, \dots, m \end{cases}$
- (3) Calculate the number of encoded data packets g of degree 1 expected in the decoding process, $g \equiv c \ln(m/\mu) \sqrt{m}$ (μ is the probability that the decoder fails to fully restore the original information; c is between 0 and 1, a certain constant)
- (4) Define a positive function $\tau(d) = \begin{cases} (g/m)(1/d) & d = 1, 2, \dots, (m/g) - 1 \\ (g/m) \log(g/\mu) & d = (m/g) \\ 0 & d > (m/g) \end{cases}$
- (5) Add the ideal soliton degree distribution ω to τ , and then normalize it to obtain a robust soliton degree distribution $\Phi(d) = (\omega(d) + \tau(d)/z)$, $Z = \sum_d (\omega(d) + \tau(d))$; $\Phi(d)$ represents the probability that the encoding packet degree is d when encoding with a robust soliton degree distribution
- (6) From the $m + 1$ th round, according to the degree distribution function, select the encoded packet with degree d to be sent
- (7) Simple copy and forward operations between nodes

ALGORITHM 1: DTRS coding strategy.

initialization, the number of packets that the node can get from the neighbor node; and $(M - v_{\min}/N)$ indicates the average number of data packets that each node can obtain from the source node in the $M - v_{\min}$ round. Therefore, the sum of these three components is the expected value of the number of packets a node obtains from the source node and neighbor nodes.

The expected number of data packets that need to be obtained after node decoding is completed. Suppose that the current node has decoded r data symbols, the total number of data symbols is N , and the node receives a codeword of degree d ; then, the probability that the node can decode new data symbols is

$$p_{r,d} = \frac{\binom{r-1}{N} (N-r)}{\binom{N}{d}}, \quad (2)$$

where $p_{r,d}$ represents the probability that a node with a codeword degree d can decode a piece of metadata when a node already has r decoded codewords. According to $p_{r,d}$, in order to enable the node to recover all the codewords as quickly as possible, the degree of the codeword changes with time as follows: the degree of the codeword required to recover Re_1 metadata pieces is not greater than 1, the degree of the codeword required to recover Re_2 metadata pieces is not greater than 2, ..., and the degree of the codeword required to recover Re_i metadata pieces is not greater than i .

$$\begin{aligned} Re_1 &= \frac{N-1}{2}, \\ Re_2 &= \frac{2N-1}{3}, \dots, \\ Re_i &= \frac{iN-1}{i+1} \forall_{i \in [1, N-1]}. \end{aligned} \quad (3)$$

To recover Re_1 metadata pieces, the mathematical expectation of the number of codewords required is K_1 , and the mathematical expectation of the number of codewords

required to recover Re_2 metadata pieces is K_2 . Therefore, the mathematical expectation to recover the number of codewords required for Re_i metadata pieces is K_i .

$$K_1 = \sum_{i=0}^{Re_1-1} \frac{N}{N-i}, K_j = K_{j-1} + \sum_{i=Re_{j-1}}^{Re_j-1} \frac{\binom{N}{j}}{\binom{i}{j-1} (N-i)}. \quad (4)$$

According to the above derivation, suppose that $W_2 = K_j + \epsilon$ ($\epsilon < K_{j+1} - K_j$) is the expectation of the number of data packets that the node needs to receive. The total number of pieces of data that the slave node can receive is expected to be W_1 , and the number of pieces of data that need to be received to complete decoding all data is W_2 ; if exactly $W_1 = W_2$, then

$$1 + 2(M - v_{\min}) + \frac{M - v_{\min}}{N} = K_j + \epsilon. \quad (5)$$

Thus, we can get the expected M of the source node to send rounds as

$$M = \frac{N(K_j + \epsilon + 2v_{\min} - 1) + v_{\min}}{2N + 1}. \quad (6)$$

According to the above two points, this paper designs the following random codeword exchange strategy, as shown in Figure 3. When a node R_i wants to exchange codeword with a randomly selected neighbor node R_j , it randomly selects a codeword s from the stored codewords and exchanges randomly selected codeword s' with a neighboring node R_j . If the codeword needs to be encoded, that is, the codeword degree is less than the maximum allowable codeword degree, and the codeword does not contain F_i , then the decoded data F_i in the local node R_i and the codeword s are encoded. See Algorithm 2 for a detailed description.

(2) *Analysis of results.* In this section, the system described above was tested. The source node sends $m = 100$ metadata pieces to $N = 10$ nodes. Figure 4 shows a

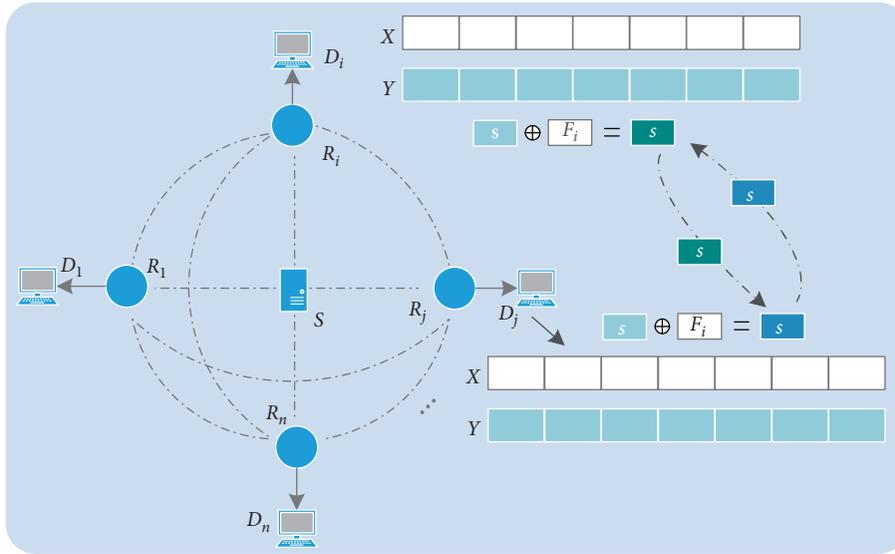


FIGURE 3: Node R_i exchanges codewords with neighbor node R_j .

- (1) The source node S selects the data to be sent according to DTRS
- (2) Initialize an empty set of decoded data units X and an empty set of complex codewords Y on each forwarding layer node $R_i (i = 1, 2, \dots, N)$
- (3) According to the principle of making all forwarding layer nodes obtain a metadata piece in the shortest round, after each node $R_i (i = 1, 2, \dots, N)$ has a metadata piece, the coding forwarding strategy begins
- (4) Set $\text{max degree} = 1$; /*the degree of all codewords transmitted between nodes in the forwarding layer cannot be greater than max degree */
- (5) **At round k** , node R_i completes the following process:
 - (1) Randomly select a codeword s from set Y
 - (2) Randomly select a decoded piece of data F_i from set X
 - (3) If $\text{degree}(s) < \text{max degree}$ & $F_i \notin s$, then encode the codeword $s = s \oplus F_i$
 - (4) If $k > K_{\text{max degree}}$, $\text{max degree} + +$; /* $K_{\text{max degree}}$ is the degree conversion time of the codeword */
 - (5) Randomly select a neighbor node R_j , and exchange the codeword s with the codeword s' randomly selected by node R_j
 - (6) Add codeword s' to set Y
 - (7) Run the iterative decoder program to decode the received codeword

ALGORITHM 2: CDCP encoding forwarding protocol.

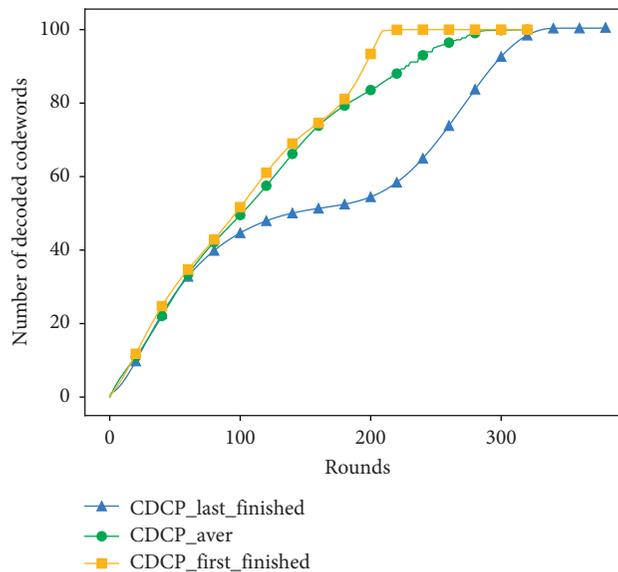


FIGURE 4: The decoding curve of the receiving node is completed first; the decoding curve of the receiving node and the average decoding curve of 10 nodes are finally completed.

comparison between the first and the last to complete the decoding curve of the receiving node and the average decoding curve of 10 nodes. It can be seen from the figure that the degree of time conversion sequence has the following problems:

- (i) Judging from the average decoding curve, we found that the postdecoding rate gradually decreased, and there was a problem of collecting “coupons.” At the last moment, the nodes received mostly redundant codewords, and some innovative codewords could not be obtained, resulting in a reduction in the overall data distribution efficiency.
- (ii) Analyzing and comparing the three decoding curves, we found that the difference between the rounds required to complete the first node and the final node is about 120 rounds. This indicates that the sequence at this time is no longer applicable to this system. There will be the slowest receiving node receiving a large number of high-degree codewords in the case of recovering a small amount of metadata pieces, greatly reducing the decoding rate.

4. Design of Codeword Distance Priority Forwarding Protocol Based on Buffer Map

4.1. Specific Design. The data exchange protocol CDCP based on codeword degree control has ensured the reliability problem of data acquisition and improved the efficiency of data distribution. However, due to the implementation of code forwarding by virtue of random data exchange and replication strategies, nodes at the forwarding layer often receive invalid coded data, which affects the data collection performance of the protocol. This section developed a dynamic adjustment forwarding strategy based on Buffer Map. Each peer in the network can use the corresponding Buffer Map to perform intelligent network coding operations, so that nodes can decode valid data from the received codewords, thus improving data distribution efficiency.

4.1.1. Encoding Package Selection Design. The peer nodes can obtain the decoding information of all nodes through Buffer Map exchange. One of the most important algorithms and strategies for file transfer between nodes is the neighbor’s Buffer Map information.

In fact, the periodic exchange of Buffer Map information requires a certain amount of bandwidth and computing resource overhead. It is assumed that each round of node requests has Buffer Map information, but the probability of the node being requested in different rounds is different. The time of Buffer Map exchange and the time of the transmission strategy made by the node are not necessarily synchronized, which may cause the Buffer Map information to expire. Therefore, it is necessary to control the frequency of Buffer Map updates to reduce the overhead and, on this basis, design a reasonable coding scheduling strategy to reduce the probability of sending useless codewords to the other party due to the expiration of Buffer Map information.

According to CDCP, the node can immediately decode a new metadata piece when the codeword distance of the packet it receives is 1. As a receiving node, the codeword distance between the received data packet and its decoded codeword should be as small as possible. The best case is that the received packets have a codeword distance of 1. However, as not every node’s Buffer Map is updated in real time, the Buffer Map may expire. The node can only select reasonable packets based on the expired Buffer Map and send them to neighboring nodes. From this perspective, after several rounds, the node may need to send data packets with a codeword distance greater than 1 to increase the probability that the packets are valid.

This paper proposed the CDPPBM, a Codeword Distance Priority Protocol based on Buffer Map, which makes full use of the expired Buffer Map to find the optimal codeword distance sequence and improve the efficiency of data distribution.

Definition 1. u is waiting for the round of Buffer Map update.

As shown in Figure 5, Buffer_map_B represents the decoded information of node B stored in node A, and X_B represents the decoded information of node B in the current round. In round u , node A selects a data packet s according to Buffer_map_B. The codeword distance between packet s and Buffer_map_B is $\text{dist}_{\text{guess}} = \text{dist}(s, \text{Buffer_map_B})$. The codeword distance between the packet s and X_B is $\text{dist}_{\text{real}} = \text{dist}(s, X_B)$. According to the previous analysis, if $\text{dist}_{\text{real}} = 1$, which means that the strategy has selected the appropriate $\text{dist}_{\text{guess}}$, node B will receive a valid codeword.

Node A sends data to node B in round u . $u = 0$ means that node A has received the Buffer Map exchange of neighboring node B in this round, and the peer node returns the data packet of $\text{dist}_{\text{guess}} = 1$. $u > 0$ means that node A in this round fails to receive the Buffer Map exchange of neighboring node B, answer the request, and return the packet with $\text{dist}_{\text{guess}} = f(u)$. Node A waits until the next time it receives the neighboring node B’s Buffer Map exchange, and u starts counting from 0 again.

4.1.2. Codeword Distance Selection Design. In the previous section, it was proposed that when $u > 0$ and $\text{dist}_{\text{guess}} = f(u)$, the specific expression of $f(u)$ is not explicitly given. Therefore, this section mainly chooses the codeword distance of the sent data packets.

Definition 2. dist is the codeword distance required by neighbor node B.

According to the expired Buffer Map, in the u th round, node A selects the encoding package Pa_{dist} with codeword distance of dist and sends it to node B. In this u , node B is constantly exchanging data with other neighboring nodes. Therefore, node B may obtain some of the dist metadata pieces from other neighboring nodes, which will facilitate the decoding of Pa_{dist} .

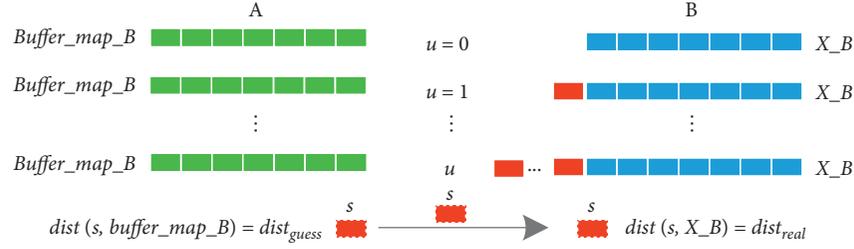


FIGURE 5: Choosing the appropriate codeword distance $\text{dist}_{\text{guess}}$ according to Buffer Map.

Definition 3. $\text{dist} \leq (j/q)$, the degree of the sent data packet cannot be greater than $(1/q)$ of the current maximum degree j , q is a positive integer.

If the randomly selected data packet is a high-degree codeword packet that reaches the optimal codeword distance, the receiving node needs to iteratively decode the high-degree codeword packet for multiple times, resulting in excessive decoding overhead. Therefore, in order to avoid sending too many high-degree packets, $\text{dist} \leq (j/q)$ is required.

Definition 4. The number of invalid data packets is expected to be E_{useless} .

If the dist metadata pieces selected by node A is obtained by neighboring node B from other neighboring nodes in u , P_{dist} is invalid for neighboring node B, and the number of invalid packets is expected to be E_{useless} :

$$E_{\text{useless}} = \left(\left(\frac{1}{f} * 1 + \frac{f-1}{f} * \delta + \frac{1}{N} * \eta \right) * \sigma * u \right) \text{dist} \quad (7)$$

Among them, f is the interval at which the Buffer Map is sent; $1/f$ means the frequency of changing neighbor node to send the Buffer Map; $(f-1/f) * \delta$ is the probability that the packet can be decoded when the Buffer Map is not sent; $(1/N) * \eta$ represents the probability of receiving a data packet from the source node and being able to decode a piece of metadata; σ represents the expectation of the amount of new metadata pieces recursively decoded; and $((1/f) * 1 + (f-1/f) * \delta + (1/N) * \eta) * \sigma * u$ is expressed as, after u rounds, the neighbor node decodes the expected amount of new metadata pieces.

Definition 5. r_0 is the amount of decoded metadata pieces of neighbor nodes.

According to the Buffer Map, we can know the amount of decoded metadata pieces of neighbor nodes. If the dist metadata pieces is taken from its undecoded set $m - r_0$, the total possible amount of dist metadata pieces is

$E_{\text{total}} = \binom{m - r_0}{\text{dist}}$. Then, the probability that the packet with the codeword distance of dist is invalid to the neighbor node is $P_{\text{useless}} = (E_{\text{useless}} / E_{\text{total}})$. Therefore, the probability that the data packet is valid for neighboring nodes is $P_{\text{use}} = 1 - P_{\text{useless}}$.

Through the above analysis, after selecting the neighboring node, the node can calculate the codeword distance dist corresponding to the maximum value of P_{use} according to the Buffer Map corresponding to the neighbor node, and this dist is the best choice at this time. Once dist is determined, the codeword is selected to be encoded and forwarded. Under the control of the conversion sequence of degree time, maximizing the P_{use} of the data to be sent can improve the effectiveness of the flowing data packets in the system and reduce the number of data packet transmissions, thus achieving the higher data distribution efficiency.

4.2. Design of Conversion Sequence of Codeword Distance Based on Genetic Algorithm. In the strategy described in the previous section, before forwarding data, a node needs to calculate P_{use} through multiple formulas to find the codeword distance dist for the maximum value of P_{use} . In this process, complex polynomial calculations will inevitably cause excessive computational overhead, affecting the efficiency of data distribution and making it outweigh the gains. To solve this problem and simplify the calculation, before the system runs, a reasonable codeword distance conversion sequence is found for the system in advance, and this sequence is used to replace the calculation process of P_{use} in the actual distribution process.

In view of the inability to theoretically derive the exact relationship between codeword distance dist and Buffer Map exchange interval f and round u , the nodes in this system are strong nodes, which can bear excessive computational costs, and heuristic algorithms such as genetic algorithm can be used to find the relationship between the codeword distance dist and Buffer Map exchange interval f and the round u , to get a codeword distance conversion sequence. After the system network is established, according to the number of nodes and file size, a genetic algorithm is run to find the optimal codeword distance conversion sequence corresponding to it. In actual operation, the system does not need to use genetic algorithm anymore, the node can directly select the codeword according to this known codeword distance sequence.

4.2.1. Nonsubtractive Proof of Codeword Distance Sequence

Theorem 1. The codeword distance dist is a nondecreasing sequence about u ; that is to say, there must be a u which makes $p(\text{dist}) < p(\text{dist} + 1)$.

Let $p(\text{dist})$ be the probability that the codeword s selected by node A according to Buffer_map_B can be decoded by node B. Let $de(u) = de_B(0) + Q$ be the number of decoded codewords of node B in the u round, where $de_B(0)$ is the length of Buffer_map_B, which is a known quantity, and Q is the node B after u round, that is, the number of new decoded codewords.

$$p(\text{dist}, de_B(u)) = \frac{\binom{de_B(u)}{\text{dist} - 1} (m - de_B(u))}{\binom{m - de_B(u)}{\text{dist}}}. \quad (8)$$

Theorem 2. *If for some $i < j$, there is $p(i, de_B(u)) < p(j, de_B(u))$, then for any $de_B(u)' > de_B(u)$, there is $p(i, de_B(u)') < p(j, de_B(u)')$.*

Proof

$$p(i, de_B(u)) < p(j, de_B(u))$$

$$\begin{aligned} &\implies \frac{(j-1)!(de_B(u) - j + 1)!}{(i-1)!(de_B(u) - i + 1)!} < \frac{(j)!(m - de_B(0) - j)!}{(i)!(m - de_B(0) - i)!} \\ &\implies \frac{(de_B(u) - j + 1)!}{(de_B(u) - i + 1)!} < \frac{(i-1)!(j)!(m - de_B(0) - j)!}{(j-1)!(i)!(m - de_B(0) - i)!}. \end{aligned} \quad (9)$$

It will now be proved by the counterevidence law. Suppose that for $de_B(u)' > de_B(u)$, there is $p(i, de_B(u)') > p(j, de_B(u)')$.

$$\begin{aligned} p(i, de_B(u)') &> p(j, de_B(u)') \\ &\implies \frac{(j-1)!(de_B(u)' - j + 1)!}{(i-1)!(de_B(u)' - i + 1)!} > \frac{(j)!(m - de_B(0) - j)!}{(i)!(m - de_B(0) - i)!} \\ &\implies \frac{(de_B(u)' - j + 1)!}{(de_B(u)' - i + 1)!} > \frac{(i-1)!(j)!(m - de_B(0) - j)!}{(j-1)!(i)!(m - de_B(0) - i)!}. \end{aligned} \quad (10)$$

Run simultaneously inequalities (9) and (10), and the results are as follows:

$$\begin{aligned} &\frac{(de_B(u)' - j + 1)!}{(de_B(u)' - i + 1)!} > \frac{(i-1)!(j)!(m - de_B(0) - j)!}{(j-1)!(i)!(m - de_B(0) - i)!} > \frac{(de_B(u) - j + 1)!}{(de_B(u) - i + 1)!} \\ &\implies \frac{(de_B(u)' - j + 1)!}{(de_B(u)' - i + 1)!} > \frac{(de_B(u) - j + 1)!}{(de_B(u) - i + 1)!} \\ &\implies (de_B(u)' - j + 1) \dots (de_B(u) - j + 2) > \\ &\quad \cdot (de_B(u)' - i + 1) \dots (de_B(u) - i + 2) \\ &\implies \frac{(de_B(u)' - j + 1)}{(de_B(u)' - i + 1)} \dots \frac{(de_B(u)' - j + 2)}{(de_B(u)' - i + 2)} > 1. \end{aligned} \quad (11)$$

Since each division on the left side of the inequality is less than 1, this result is clearly untenable. Therefore, it can be concluded that the initial assumption is wrong, and the theorem is proved. It can be proved that when $de_B(u)$ reaches

a certain point as u continues to increase, if $i < j$, the data packet with the codeword distance j is more likely to be decoded than the data packet with the codeword distance i . Therefore, there is a nonsubtractive sequence of codeword

distance $dist$ with respect to u . Since in $de_B(u) = de_B(0) + Q$, the Q value is an unknown quantity and may change at any time, in theory we cannot get the specific relationship between Q and u , and theoretically we cannot get the codeword. For the sequence of distance $dist$ with respect to u , this paper uses genetic algorithm to find this optimal sequence, which makes the system distribution efficiency the highest. \square

4.2.2. Genetic Algorithm Design. In the last section, it has been proved that the codeword distance conversion sequence is a nondecreasing sequence. Through the corresponding experiments in the last period, we know that the codeword distance conversion sequence is a power function related to u ,

$$dist = au^b + cu + 1. \quad (12)$$

When the file $size$ and the number of nodes N are determined, the genetic algorithm is used to optimize the coefficients a, b, c and Buffer Map transmission interval f , and the number of rounds of data packets sent by the source node is minimized.

f is an integer, and a, b , and c are all floating-point numbers. Since the binary expression of floating-point numbers cannot be directly used as genes, we can use integer binary expression to represent floating-point numbers according to the method in [34]. The chromosome is composed of the four parameters f, a, b , and c . Each parameter corresponds to a gene length of 8 bits and a total chromosome length of 32 bits.

The entire genetic algorithm mainly includes the following:

- (1) Population initialization.
- (2) Setting elite reservation.
- (3) Obtaining chromosome fitness value.
- (4) Calculating the cumulative probability of chromosome.
- (5) Crossover and mutation.
- (6) Repeating steps 3–5. Through this process, the optimal f, a, b, c can be finally obtained.

4.3. Degree of Time Conversion Sequence Optimization. If the randomly selected data packet reaches the optimal codeword distance and is a height codeword packet, then the receiving node needs to iteratively decode the height codeword packet multiple times in order to decode the innovative codeword in the coded packet. This will cause excessive decoding overhead. In order to solve this problem, we follow the ideas in the CDCP to control the degree of growth of encoded packets.

According to the theoretical derivation of the CDCP, it can be known that the probability of receiving a data packet with a degree j and a codeword distance of 1 when the current maximum degree is j and the number of decoded codewords is i , is

$$p_{i,j} = \frac{\binom{i}{j-1} (m-i)}{\binom{m}{j}}. \quad (13)$$

Among them, the denominator indicates the number of types of packets received with degree j , and the numerator indicates the number of types of packets received with degree j and codeword distance 1.

In this scheme, the data packets exchanged between the nodes are not randomly selected, but according to the Buffer Map, the data packets that are valid for the receiver (the data packets have the metadata pieces required by the receiver) are sent. Therefore, the types of data packets received by each receiving node may change. Set the codeword distance of the transmitted data packet to be no greater than $S(1/q)$ of the current maximum degree j ; that is, $dist \leq (j/q)$.

Therefore, the number of possible types of data packets received by each node is changed from

$$p'_{i,j} = \frac{\binom{i}{j-1} (m-i)}{\sum_{x=1}^{(j/q)} \binom{i}{j-x} \binom{m-i}{x}}. \quad (14)$$

The conversion point K'_j will also change with the degree:

$$K'_j = K'_{j-1} + \sum_{i=Re_{j-1}}^{Re_{j-1} + \sum_{x=1}^{(j/q)} \binom{i}{j-x} \binom{m-i}{x}} \frac{\binom{i}{j-x} \binom{m-i}{x}}{\binom{i}{j-1}}. \quad (15)$$

With the addition of the Buffer Map and codeword distance prediction strategy, the degree conversion sequence of the CDCP has been changed, which can improve the effectiveness of the encoded package to the neighboring nodes under the current maximum permissibility.

4.4. Overall Data Distribution Description. According to the forwarding strategy designed above, combined with the DTRS encoding strategy of the source node, the overall data distribution is divided into the following points. First, according to the system configuration, the appropriate Buffer Map transmission interval and codeword distance conversion sequence are selected by genetic algorithms for the system. Second, after the source node receives the file, the DTRS coding strategy is adopted for data distribution, and the CDPPBM is used for codeword coding exchange between nodes. For details, see Algorithm 3. Finally, when the node completes decoding and sends feedback to the source node, after receiving the feedback from all nodes, the source node stops the current generation of data distribution.

Initialization: refer to steps 1 to 4 of Algorithm 2

- (1) Set the sending frequency of Buffer Map to f . Assuming that the node R_i sends a Buffer Map to the neighbor node R_j this round, the node R_j responds to the request and returns the Buffer Map of the node R_j in the returned data packet to update the other party's Buffer Map.
- (2) At round k , node R_i completes the following process:
 - (1) Randomly select a neighbor node R_j from the IP list for data exchange.
 - (2) Find the Buffer Map list of the neighbor node R_j ; if it is empty, jump to 3; otherwise, if the update waiting round $u = 0$, jump to 4, and if $u > 0$, jump to 5.
 - (3) The Buffer Map list of the neighbor node R_j is empty, and a codeword s is randomly selected from the set Y ; a decoded piece of data F_i is randomly selected from the set X ; if $\text{degree}(s) < \max \text{degree}$ and $F_i \notin s$, then the codeword is encoded: $s = s \oplus F_i$; jump to 6.
 - (4) The Buffer Map list of the neighbor node R_j is not empty, and the optimal codeword distance $d_{\text{best}} = 1$ is found according to the codeword distance sequence. Take the codeword s from X , which requires $\text{dist}(s, \text{BufferMap}) = 1$; jump to 6.
 - (5) The Buffer Map list of the neighbor node R_j is not empty, and the optimal codeword distance d_{best} is found according to the codeword distance sequence. Take the codeword from X and Y and encode it into codeword s , which requires $\text{degree}(s) < \max \text{degree}$ and $\text{dist}(s, \text{BufferMap}) \leq d_{\text{best}}$; jump to 6.
 - (6) If $k > K'_{\max \text{degree}}$, $\max \text{degree} + +$; $/ * K'_{\max \text{degree}}$ is the degree conversion time of the codeword $*$.
 - (7) Exchange the codeword s with the codeword s' randomly selected by node R_j .
 - (8) Add codeword s' to set Y ; run the iterative decoder program to decode the received codeword.

ALGORITHM 3: CDPPBM encoding forwarding protocol.

4.5. Optimization Based on Data Segmentation. The above research is aimed at small files. When the file received by the source node is a large file, such as a video file, then according to the above scheme, the system will seriously affect the data distribution efficiency due to too many fragments and cumbersome scheduling. Therefore, guessing the shard size will be an important factor affecting data distribution.

4.5.1. Preliminary Experiment. In order to prove the existence of coding units for data distribution when determining file size and node number, this paper took files of different sizes (10000, 15000, and 20000 metadata pieces) for verification experiment. The encoding units were set as 16 KB, 32 KB, and 63 KB, respectively, and the system running time was taken as the measurement index. The same set of parameters were averaged through 10 experiments. As can be seen from Figure 6, with the increase of the encoding unit, the receiving completion time gradually decreases; in particular, for the file with the size of 20000, the encoding unit is 16 KB. Under the same conditions, the distribution time of 16 KB bits is about 20 times that of 63 KB bits. Therefore, it can be proved that code size is an important factor affecting data distribution. It can be inferred from the above experimental results that the larger the file is, the larger the optimal coding unit is.

4.5.2. Design of File Block Contiguous Protocol. This article uses the UDP, and according to the UDP data packet maximum capacity of 65536 bytes, we cannot use experiments to prove our guess. Therefore, this paper proposes a block concatenation protocol, BPP. According to the characteristics of the UDP packet capacity, for the large file F , set the encoding unit to 63 KB (the maximum UDP packet is 64 KB, and the remaining 1 KB needs to be reserved for other information such as IP header), and the m -block encoding unit is combined into a slice F_{piece} ; there are $n =$

$(|F|/m * 63\text{KB})$ pieces. The encoding between each piece is independent. The source node sends them in order according to the order of the pieces. The data distribution method of each piece follows the CDPPBM strategy.

Assuming that the time required for the system to distribute a piece is τ' , the total time required to distribute the file F is $T = n * \tau'$. The greater the number of coding unit blocks m contained in a slice, the longer the time τ' for distributing a slice, but the smaller the number of fragments n , the less the number of transmissions. Therefore, there is a balance between n and τ' to minimize the total transmission time T . The smaller number of segments can make file segment scheduling easier, thereby improving the efficiency of file distribution, but the lower number of segments also means that the speed of encoding and decoding decreases, and a balance needs to be achieved between the two.

When knowing the number of nodes and file size, in order to find the optimal fragment size, this section will add the fragment size as a gene to the chromosome according to the design of the previous section. After multiple generations, the optimal segment size is found. The specific process is shown in Figure 7.

5. Experiment

In order to evaluate the performance of the protocol proposed in this paper, the RSD, DTRS, CDCP, and CDPPBM protocols are implemented, respectively. Each protocol performs multiple sets of comparative experiments on different sizes of m and N to verify the data distribution performance of the protocol proposed in this paper. The setting of m and N is based on the number of receivers and the number of messages distributed in each generation in the actual system. On the other hand, for large files, determine the number of nodes, experiment with files of different sizes, and select the optimal data segmentation method for the files.

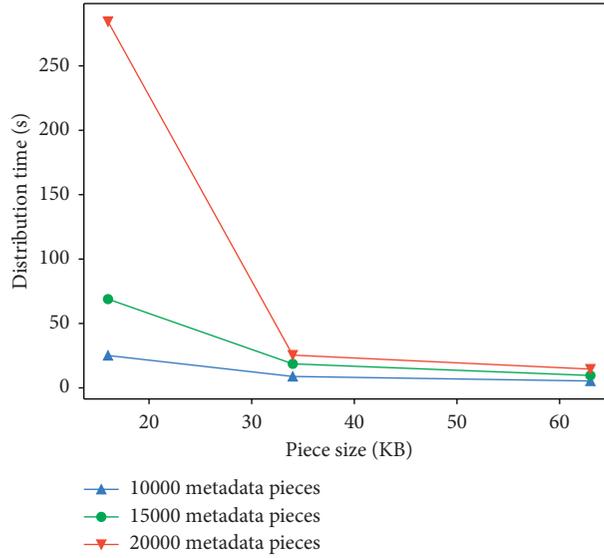


FIGURE 6: Comparison of data distribution time of files of different sizes in different coding units.

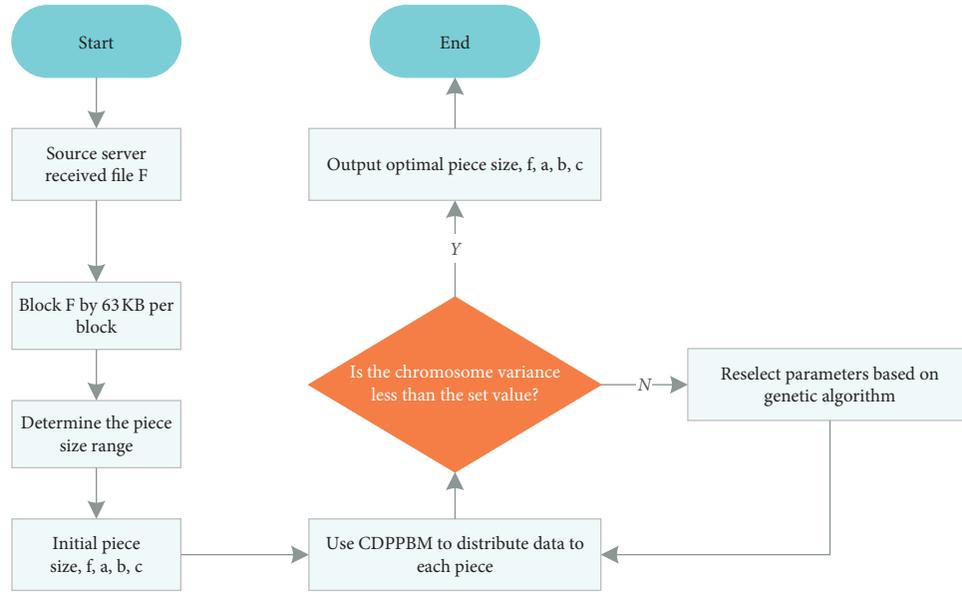


FIGURE 7: The process of selecting the optimal shard size.

5.1. *Experimental Parameters and Indicators.* The experiment was implemented by about 2000 lines of Python code. We will conduct experiments on servers under the same network. The experiment consists of a source node and 15 receiving nodes with the same configuration. Source node specific configuration: CPU architecture is x86_64, 8 cores; memory 13869 MB; hard disk 300 GB. Node configuration: CPU architecture is x86_64, 2 cores; memory 8196 MB; hard disk 40 GB. Each node is a strong node, with large storage capacity and strong computing power. The transmission file size m ranges between 30 and 150 metadata pieces.

In order to better evaluate the strategy proposed in this paper, we will start with the following performance indicators:

- (i) Source node sending round: since the main purpose of our protocol is to reduce the pressure on the source node, the source node sending round becomes the main performance evaluation indicator.
- (ii) System running time: when the transmitted data packet carries Buffer Map information, the total length of the data packet changes, which will have a certain impact on the transmission time. Therefore, when considering the rounds sent by the source node, the total running time of the system also needs to be considered.
- (iii) Node average decoding rate: it can directly reflect the receiving status of the node. If the source node can make all nodes complete the reception in fewer

rounds, this means that the average decoding rate of the protocol is higher.

- (iv) Node decoding completion difference: in the case of multiple receiving nodes, it is unavoidable to complete the reception time in sequence, so the difference can reflect the stability of the protocol.
- (v) Data distribution efficiency improvement rate: take the original system source node sending round $\text{Round}_{\text{RSD}}$ as the basis. According to the following formula, the improvement agreement $\text{Round}_{\text{improve}}$ relative to the original system agreement data distribution efficiency increase rate $\text{distribute}_{\text{raise}}$ can be calculated.

$$\text{distribute}_{\text{raise}} = \frac{\text{Round}_{\text{RSD}} - \text{Round}_{\text{improve}}}{\text{Round}_{\text{RSD}}} \quad (16)$$

The greater $\text{distribute}_{\text{raise}}$ means that the protocol data distribution efficiency is higher.

5.2. Codeword Distance Priority Experiment Based on Buffer Map

5.2.1. Comparison of Data Distribution Efficiency of Various Protocols. In order to compare the data distribution efficiency of each protocol, we use each protocol to send the selected $m = 100$ metadata pieces to $N = 10$ nodes. The comparisons of source node sending rounds and node decoding efficiency for each protocol are shown in Figure 8. As shown in the figure, the number of the rounds required for all nodes of the CDPPBM to complete the reception is far less than that of the RSD (690 rounds), about 142 rounds, implying that the pressure on the source node can be reduced greatly. In the RSD strategy, due to the cliff effect, the accumulation of high-degree codewords in the early stage makes the decoding efficiency very low, and efficient decoding starts only after enough 1° codewords are received in the later period.

From the point of view of node decoding efficiency, the CDPPBM, as an effective solution to the problem of collecting coupons in DTRS and CDCP, still maintains a high decoding efficiency at the last moment. CDPPBM can realize the codewords required by neighboring nodes, which helps to avoid multiple forwarding of invalid codewords and effectively alleviate the problem that the last few codewords cannot be collected.

5.2.2. Comparison of Data Distribution Efficiency of Different Protocols under Different Conditions. In order to determine the effect of file size and number of nodes on each protocol, this paper takes different m and N for comparative experiments, where $m = 30, 100, 150$ and $N = 5, 10, 15$. Each combination is tested multiple times on average. The experimental results are shown in Figure 9, where 9(a)–9(c) and 9(d)–9(f), respectively, represent the comparison of sending rounds and running time required by the source

node when different nodes receive 30, 100, and 150 metadata pieces under various protocols. From the above figure, we draw following conclusions: As shown in Figure 9(c), when the number of nodes increases and the files are relatively large, CDCP cannot improve the efficiency of data distribution due to the random selection of data for sending, making it not much better than DTRS protocol. It can be seen from the figure above that no matter which combination of CDPPBM is used, the number of sending rounds of the source node is far less than that of the RSD, and all have good performance.

5.2.3. Comparison of Data Distribution Efficiency Improvement Rate. According to the data distribution efficiency improvement rate defined in (16), the value is calculated in a combination of various conditions, and the calculation results are shown in Tables 2–4.

As the number of nodes under different size files increases, $\text{distribute}_{\text{raise}}$ has significantly improved. The CDPPBM has a higher $\text{distribute}_{\text{raise}}$, and $\text{distribute}_{\text{raise}}$ is significantly higher than other protocols in all combinations. From the above tables, it can be concluded that the file size has no obvious effect on $\text{distribute}_{\text{raise}}$ with the utilization of CDPPBM, indicating that the protocol may have good performance in various file sizes.

5.2.4. Node Decoding Rate Comparison of Different Protocols under Different Conditions. The node decoding rate can directly reflect the receiving situation of the node. In order to determine the impact of the file size and the number of nodes on each protocol, this paper experimented with three files with sizes of 30, 100, and 150 pieces of metadata pieces. The number of receiving nodes is divided into three cases of 5, 10, and 15. The four protocols are tested in different combinations of file size and number of nodes, with a total of 36 combinations, each of which is averaged multiple times.

As shown in Figure 10, (a) $m = 30, N = 5$, (b) $m = 30, N = 10$, (c) $m = 30, N = 15$, (d) $m = 100, N = 5$, (e) $m = 100, N = 10$, (f) $m = 100, N = 15$, (g) $m = 150, N = 5$, (h) $m = 150, N = 10$, (i) $m = 150, N = 15$. We can see the shortcomings of RSD from each decoding curve. A large number of high-degree codewords are received in the initial stage. Although it can cover a large amount of source data, the decoding algorithm cannot be triggered immediately due to the lack of 1° packets, and the decoding rate in the initial stage is very low. In the later stage, the decoding rate increased significantly, causing the “cliff effect.” In the DTRS protocol, because the node does not perform the encoding operation and randomly selects data packets for transmission under the uncontrolled condition, the node may receive a large number of data packets with high codewords that cannot be decoded immediately, and there is a serious problem of collecting “coupons.” The time conversion sequence in the CDCP effectively alleviates the problem of high codeword and further improves the decoding efficiency. In the CDPPBM, the Buffer Map and codeword distance priority strategies are added to effectively solve the problem of

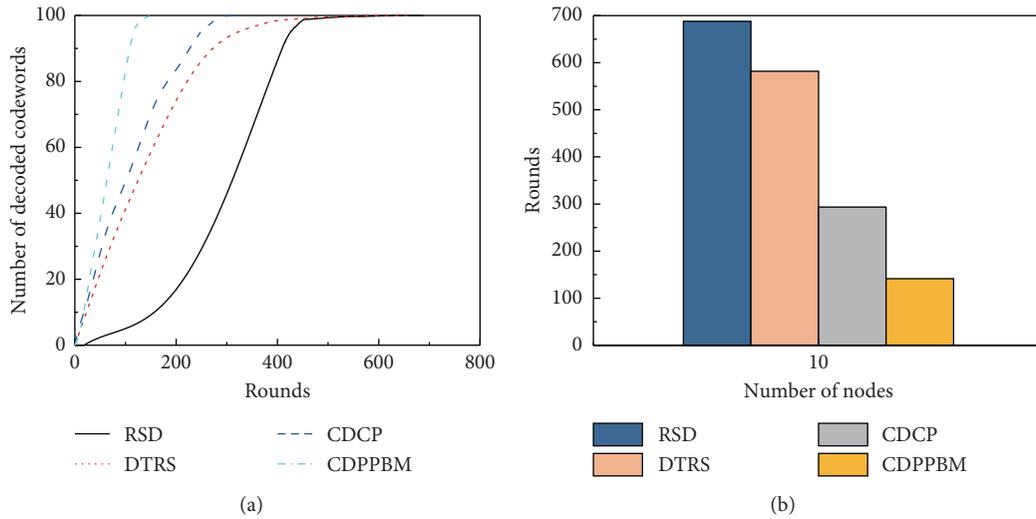


FIGURE 8: Comparison of source node sending rounds and node decoding efficiency of each protocol.

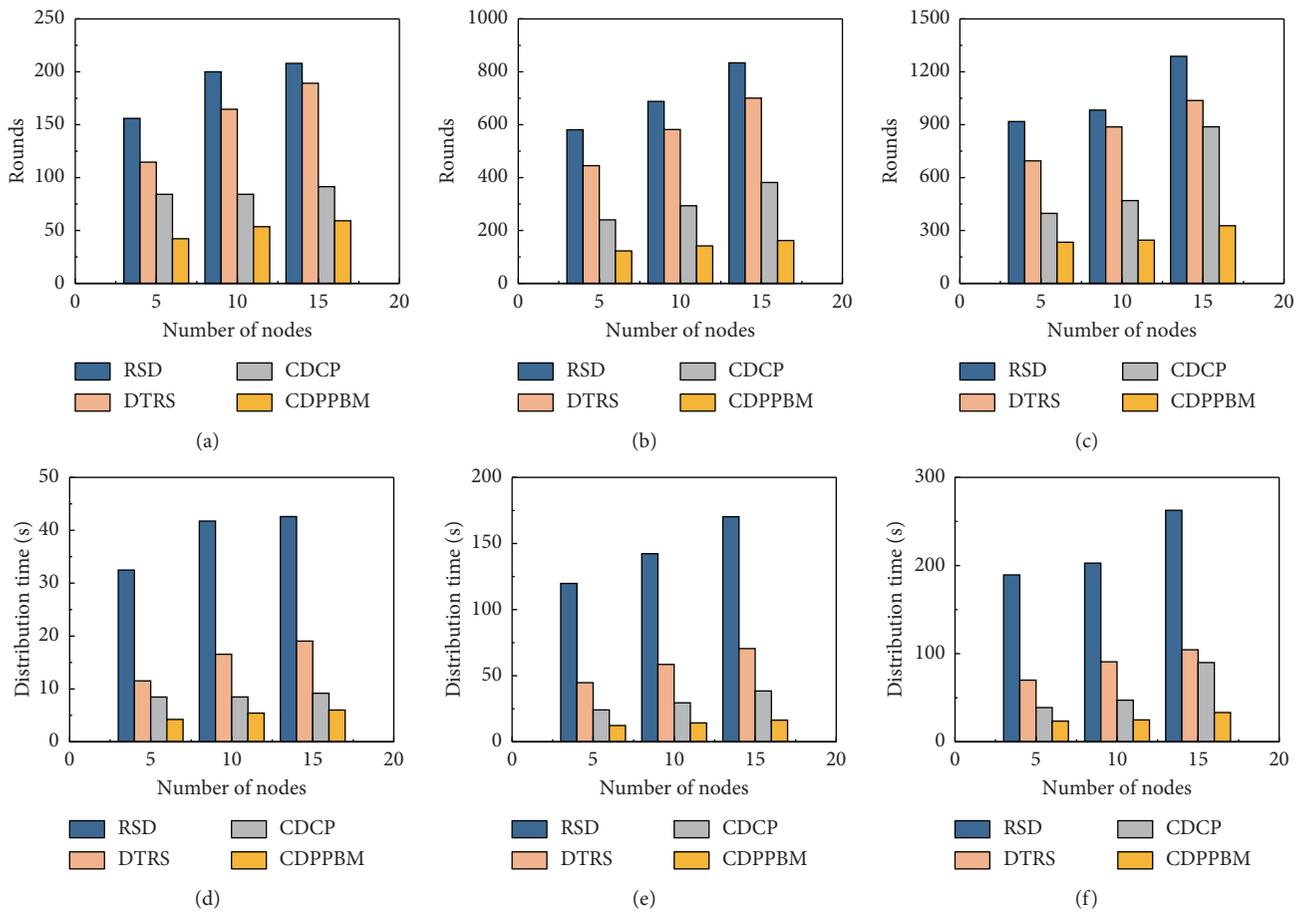


FIGURE 9: Comparison of the rounds and running time sent by the source node under different conditions.

TABLE 2: $m = 30$, corresponding to the number of different nodes in different protocols distribute_{raise}.

Number of nodes	DTRS (%)	CDCP (%)	CDPPBM (%)
5	26.54	46.03	72.95
10	17.7	57.9	73.2
15	9.04	56.06	71.54

TABLE 3: $m = 100$, corresponding to the number of different nodes in different protocols distribute_{raise}.

Number of nodes	DTRS (%)	CDCP (%)	CDPPBM (%)
5	23.41	58.66	78.86
10	15.44	57.33	79.42
15	16.02	54.24	80.58

TABLE 4: $m = 150$, corresponding to the number of different nodes in different protocols distribute_{raise}.

Number of nodes	DTRS (%)	CDCP (%)	CDPPBM (%)
5	24.34	56.75	74.58
10	9.73	52.21	75.04
15	19.49	31.07	74.61

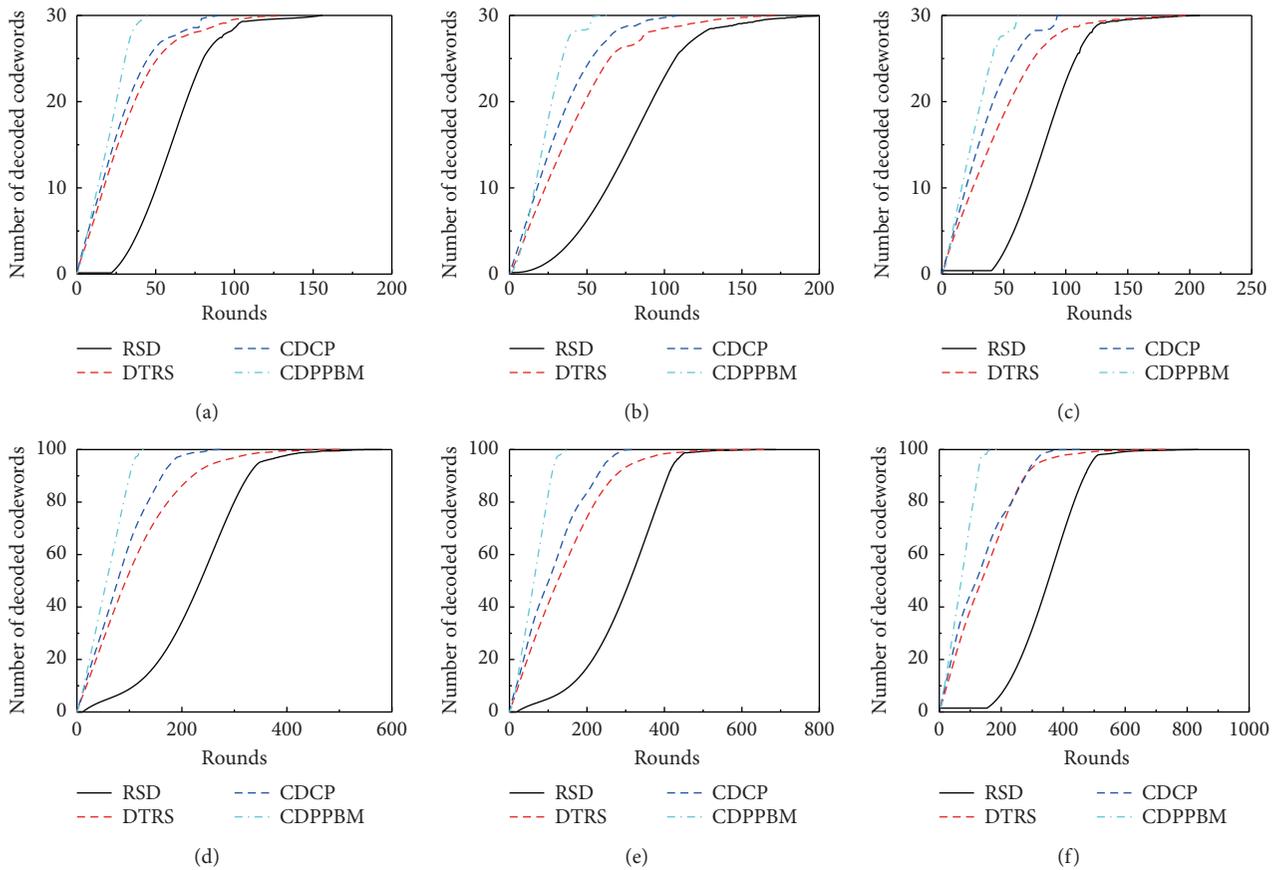


FIGURE 10: Continued.

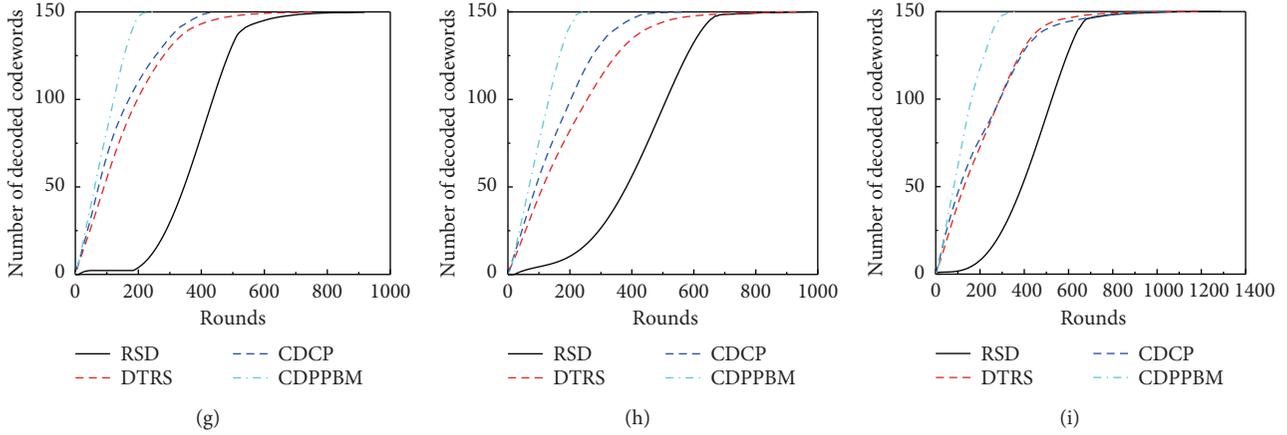


FIGURE 10: Comparison of node decoding under different conditions.

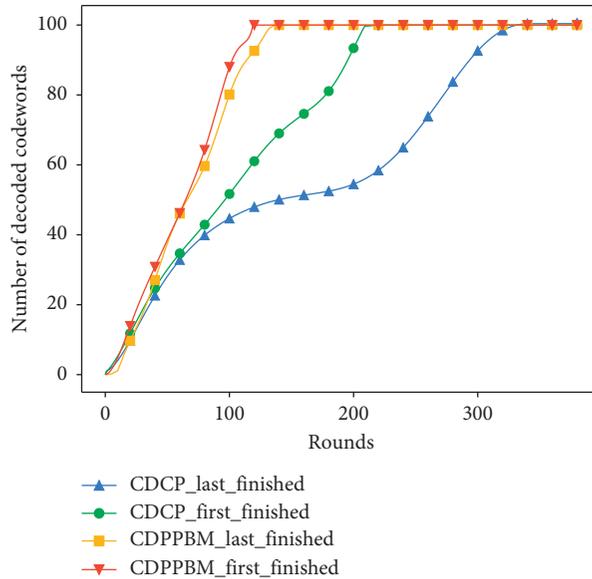


FIGURE 11: Comparison of the decoding rate of the fastest and slowest nodes of the two protocols.

coupon collection and greatly improve the decoding efficiency of the node and the speed of the data distribution of the system.

5.2.5. Extreme Delay Rate Optimization. The utilization of the CDCP-based time conversion sequence method will cause problems of low last decoding rate and large difference in decoding completion time between nodes in the system. The system running time is determined by the completion time received by the last node. If a certain protocol can not only make the last round of reception completion of the last node smaller, but also ensure that the round difference between the first reception completion node and the last completion reception node is smaller, this implies that the data distribution protocol is efficient. Therefore, the CDCP and CDPPBM are used to send 100 selected pieces of metadata pieces to 10 nodes for extreme delay rate comparison. The data with the fastest and the slowest decoding completion in the experimental

results are taken out and plotted, as shown in Figure 11. As far as the extreme delay rate is concerned, it can be seen from the figure that the fastest decoding completion curve greatly differs from the slowest decoding completion curve of the CDCP in the completion time point, and the difference of about 150 rounds indicates that the fastest decoding completion node is almost half of the wait time.

5.2.6. Optimization Experiment Based on Data Segmentation. According to the previous analysis, we know that after determining the file size and the number of nodes, we can find a balance between encoding overhead and scheduling overhead, that is, the optimal fragment size, making the CDPPBM optimize the efficiency of data distribution under this condition.

Experiments are conducted on files with sizes of 4032 KB (64 blocks), 8064 KB (128 blocks), 16128 KB (256 blocks), and 32256 KB (512 blocks), and the optimal fragment size is

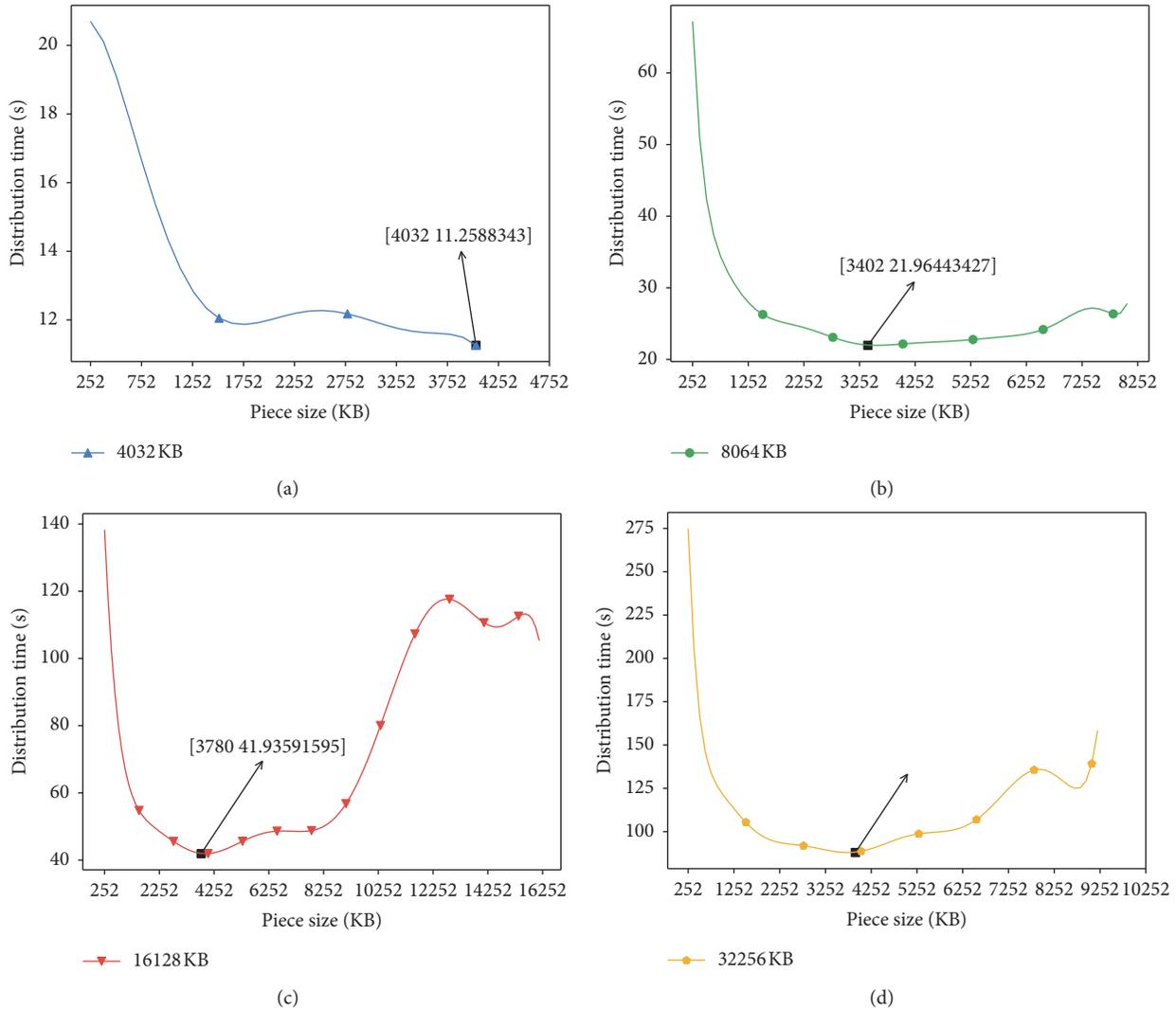


FIGURE 12: Comparison of data distribution time of files of different sizes under different fragment sizes.

selected for each file. First, the genetic algorithm is utilized to find the optimal fragment size, Buffer Map transmission interval, and codeword distance sequence. Then, set the Buffer Map transmission interval and codeword distance sequence, change the fragment size, and get the data distribution time corresponding to different fragment sizes.

It can be observed from Figure 12 that there is a balance point for files of different sizes to find that the data distribution efficiency is the highest. Moreover, considering the influence of other factors of the system, the optimal fragment size may be an interval within which the data distribution efficiency is close to the optimal value found by the genetic algorithm. Besides, the optimal fragment size increases as the file size increases.

6. Conclusion

In order to solve the problem of efficiently distributing large-scale, high-concurrency, and continuous fishing vessel positioning information in the actual system, this paper

proposes two strategies, CDPPBM and BPP. Firstly, on the basis of an analysis of the reasons for the low efficiency of data distribution in the original system, a network coding data distribution model based on UDP is proposed. In this model, network coding is used to ensure reliable data distribution and improve data distribution efficiency. Different encoding methods are used between the source node and the node to reduce the pressure on the source node, so that the source node and the node achieve load balancing. In order to further improve the efficiency of data distribution and increase the concentration of innovative codewords in the network, CDPPBM is proposed. This strategy can enable the node to obtain the required codeword faster, improve the decoding efficiency of the node, and to a certain extent alleviate the final “coupon” collection problem, so that the overall data distribution efficiency is significantly improved. For large files, the proposed BPP strategy finds the balance between system scheduling and network coding overhead, that is, selecting the optimal fragment size, so that the data distribution efficiency is the highest.

The CDPPBM and BPP data distribution protocols proposed in this paper consider only the situation of good network condition but ignore the existence of data loss. In fact, when the network condition is poor, data loss is inevitable. Therefore, in the future work, it is hoped that the corresponding algorithms can be designed under the condition of poor network to maintain high data distribution efficiency.

Data Availability

The raw/processed data required to reproduce these findings cannot be shared at this time as the data also form part of an ongoing study.

Conflicts of Interest

The authors declare that there are no conflicts of interest.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (no. 61772165) and the China Postdoctoral Science Foundation under Grant no. 2016M600465.

References

- [1] T. Pilar-Fonseca, A. Campos, P. Fonseca et al., "The importance of satellite-based vessel monitoring system (vms) for fisheries management: a case study in the Portuguese trawl fleet," in *Maritime Engineering and Technology*, p. 19, CRC Press, Boca Raton, FL, USA, 2012.
- [2] A. Magonetto, R. Gaeta, M. Grangetto, and M. Sereno, "P2p streaming with lt codes: a prototype experimentation," in *Proceedings of the 2010 ACM Workshop on Advanced Video Streaming Techniques for Peer-To-Peer Networks and Social Networking*, pp. 7–12, October 2010.
- [3] M. Luby, "Lt codes," in *Proceedings of the 43rd Annual IEEE Symposium on Foundations of Computer Science, 2002*, pp. 271–280, IEEE, Vancouver, Canada, November 2002.
- [4] Y.-h. Chu, S. G. Rao, and H. Zhang, "A case for end system multicast (keynote address)," *ACM SIGMETRICS Performance Evaluation Review*, vol. 28, no. 1, pp. 1–12, 2000.
- [5] J. W. Byers, M. Luby, M. Mitzenmacher, and A. Rege, "A digital fountain approach to reliable distribution of bulk data," *ACM SIGCOMM Computer Communication Review*, vol. 28, no. 4, pp. 56–67, 1998.
- [6] A. Kamra, V. Misra, J. Feldman, and D. Rubenstein, "Growth codes," *ACM SIGCOMM Computer Communication Review*, vol. 36, no. 4, pp. 255–266, 2006.
- [7] 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, 2018.
- [8] J. Broch, D. A. Maltz, D. B. Johnson, Y. C. Hu, and J. Jetcheva, "A performance comparison of multi-hop wireless ad hoc network routing protocols," in *Proceedings of the 4th Annual ACM/IEEE International Conference on Mobile Computing and Networking*, pp. 85–97, Dallas, TX, USA, October 1998.
- [9] H. Hosseinmardi and F. Lahouti, "Online multicasting using network coding in energy constrained wireless ad hoc networks," in *Proceedings of the 2008 3rd International Symposium on Wireless Pervasive Computing*, pp. 545–549, IEEE, Santorini, Greece, May 2008.
- [10] X. Jiao, X. Wang, and X. Zhou, "Active network coding based high-throughput optimizing routing for wireless ad hoc networks," in *Proceedings of the 2008 4th International Conference on Wireless Communications, Networking and Mobile Computing*, pp. 1–5, IEEE, Dalian, China, October 2008.
- [11] Y. Wu, P. A. Chou, and S.-Y. Kung, "Minimum-energy multicast in mobile ad hoc networks using network coding," *IEEE Transactions on Communications*, vol. 53, no. 11, pp. 1906–1918, 2005.
- [12] H. Li, H. Yu, L. Li, and A. Liu, "On the cascading data collection mechanism in wireless sensor networks," in *Proceedings of the 2007 International Conference on Wireless Communications, Networking and Mobile Computing*, pp. 2479–2482, IEEE, Shanghai, China, September 2007.
- [13] X. Tang and J. Xu, "Adaptive data collection strategies for lifetime-constrained wireless sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 19, no. 6, pp. 721–734, 2008.
- [14] J. Wang, G. Yang, X. Lin, Z. He, and J. Lin, "Continuous data collection in wireless sensor networks through pnc and distributed storage," in *Proceedings of the 2007 International Conference on Wireless Communications, Networking and Mobile Computing*, pp. 2568–2571, IEEE, Shanghai, China, September 2007.
- [15] S. Wan, Y. Zhao, T. Wang, Z. Gu, Q. H. Abbasi, and K.-K. R. Choo, "Multi-dimensional data indexing and range query processing via voronoi diagram for internet of things," *Future Generation Computer Systems*, vol. 91, pp. 382–391, 2019.
- [16] L. Li, F. Zhu, H. Sun, Y. Hu, Y. Yang, and D. Jin, "Multi-source information fusion and deep-learning-based characteristics measurement for exploring the effects of peer engagement on stock price synchronicity," *Information Fusion*, vol. 69, pp. 1–21, 2021.
- [17] C. S. Gkant and P. Rodriguez, "Network coding for large scale content distribution," in *Proceedings of the IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies*, p. 2, Miami, FL, USA, March 2005.
- [18] C. Gkantsidis and P. R. Rodriguez, "Network coding for large scale content distribution," in *Proceedings IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies*, vol. 4, pp. 2235–2245, IEEE, 2005.
- [19] M. Anandaraj, K. Selvaraj, P. Ganeshkumar, and K. Vijayakumar, "Intelligent packet coding-intelligent selection of packets for network code based content distribution in p2p network," in *Proceedings of the 2019 International Conference on Recent Advances in Energy-Efficient Computing and Communication (ICRAECC)*, pp. 1–7, IEEE, Nagercoil, India, March 2019.
- [20] J. Xu, J. Zhao, X. Wang, and X. Xue, "Swifter: chunked network coding for peer-to-peer content distribution," in *Proceedings of the 2008 IEEE International Conference on Communications*, pp. 5603–5608, IEEE, Beijing, China, May 2008.
- [21] S. Wan, J. Hu, C. Chen, A. Jolfaei, S. Mumtaz, and Q. Pei, "Fair-hierarchical scheduling for diversified services in space, air and ground for 6g-dense internet of things," *IEEE Transactions on Network Science and Engineering*, In press, 2020.
- [22] J. Lu, Y. Xin, Z. Zhang, S. Tang, C. Tang, and S. Wan, "Extortion and cooperation in rating protocol design for

- competitive crowdsourcing,” *IEEE Transactions on Computational Social Systems*, vol. 8, no. 1, pp. 246–259, 2021.
- [23] M. Wang and B. Li, “How practical is network coding?” in *Proceedings of the 2006 14th IEEE International Workshop on Quality of Service*, pp. 274–278, IEEE, New Haven, CT, USA, June 2006.
- [24] D. M. Chiu, R. W. Yeung, J. Huang, and B. Fan, “Can network coding help in p2p networks,” in *Proceedings of the 2006 4th International Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks*, pp. 1–5, IEEE, Boston, MA, USA, March 2006.
- [25] T. Hoßfeld, K. Tutschku, and D. Schlosser, “Influence of the size of swapping entities in mobile p2p file-sharing networks,” in *Proceedings of the Kommunikation in Verteilten Systemen (KiVS), Gesellschaft für Informatik eV*, Kaiserslautern, Germany, March 2005.
- [26] M. Sirivianos, J. H. Park, X. Yang, and S. Jarecki, “Dandelion: cooperative content distribution with robust incentives,” in *Proceedings of the USENIX Annual Technical Conference*, Santa Clara, CA, USA, June 2007.
- [27] R. Sherwood, R. Braud, and B. Bhattacharjee, “Slurpie: a cooperative bulk data transfer protocol,” in *Proceedings of the IEEE INFOCOM 2004*, pp. 941–951, IEEE, Hong Kong, China, March 2004.
- [28] P. Marciniak, N. Liogkas, A. Legout, and E. Kohler, “Small is not always beautiful,” 2008, <http://arxiv.org/abs/0802.1015>.
- [29] M. Anandaraj, P. Ganeshkumar, K. Vijayakumar, and K. Selvaraj, “An efficient framework for large scale multimedia content distribution in p2p network: I2nc,” *The Scientific World Journal*, vol. 2015, Article ID 303505, 12 pages, 2015.
- [30] J. Munding, R. Weber, and G. Weiss, “Optimal scheduling of peer-to-peer file dissemination,” *Journal of Scheduling*, vol. 11, no. 2, pp. 105–120, 2008.
- [31] M. Ponc, S. Sengupta, M. Chen, J. Li, and P. A. Chou, “Multi-rate peer-to-peer video conferencing: a distributed approach using scalable coding,” in *Proceedings of the 2009 IEEE International Conference on Multimedia and Expo*, pp. 1406–1413, IEEE, New York, NY, USA, July 2009.
- [32] Y. Qiao, S. Zhang, C. Zhang et al., “Svdn: packetization and layer synchronization for scalable video delivery over peer-to-peer networks,” *Multimedia Systems*, vol. 21, no. 4, pp. 327–343, 2015.
- [33] S. Sanghavi, B. Hajek, and L. Massoulié, “Gossiping with multiple messages,” *IEEE Transactions on Information Theory*, vol. 53, no. 12, pp. 4640–4654, 2007.
- [34] M. D. Vose, *The Simple Genetic Algorithm: Foundations and Theory*, MIT Press, Cambridge, MA, USA, 1999.