Wireless Multiple Access Network With a Sectional Configured Data Collection System and Codeword Filtering Mechanism

WEI ZHANG, YUWEI HU, YONGJIAN REN, AND YANGYANG BAI

1School of Computer Science and Technology, Hangzhou Dianzi University, Zhejiang 310018, China
2Key Laboratory of Complex Systems Modeling and Simulation, Ministry of Education, Hangzhou Dianzi University, Zhejiang 310018, China
Corresponding author: Yuwei Hu (hywjjf@163.com)

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

ABSTRACT Swarm robotics is an approach to collective robotics with which it is easily possible to complete the tasks that are difficult to do with single robot. The wireless multiple access network contributes significantly in wireless mobile swarm robotics for communication and networking, especially in disaster scenarios. Since the unpredictable impacts may fail the sensing robots or nodes, causing the loss of valuable data, a coding scheme called the Growth codes is designed to increase data persistence. In the wireless multiple access network, a key requirement of data collection in disaster scenarios is to maintain a high data intermediate recovery rate. In the later period of the Growth codes, the large amount of redundant data in the network affects the efficiency of data collection. In this paper, we design and analyze techniques to reduce the number of redundant data, and propose a sectional configured data collection strategy based on the Growth codes, called the SCGC protocol. Setting cache nodes around the sink to filter the redundant codewords, the sink can collect data faster, ensuring a high data intermediate recovery rate. We also design an information update strategy and some constraints to control node overhead. Through the simulations, we show that the proportion of redundant codewords can be reduced by 10-15%, while the negative impact of these valid codewords is not considered in the Growth codes. We also show that the SCGC protocol can improve the data recovery efficiency without affecting the stability of the network or shortening the network lifetime.

INDEX TERMS Codeword filtering, data recovery efficiency, multiple access, nodes scheduling, swarm robotics, wireless network, wireless mobile robot swarms.

I. INTRODUCTION

The wireless sensor network (WSN) is used on a large scale to monitor real-time environmental status that requires minimum human intervention [1]. Recently, the wireless multiple access network has gained significant importance in industrial automation [2], robot control [3] and swarm mobile robotics. Swarm mobile robotics is an approach to collective mobile robotics with which it is easily possible to complete the tasks that are difficult to do with single robot. Similar to the sensor nodes in WSN, the coordination among mobile sensing robots is based on the data collection and communication. Application areas of swarm mobile robotics are environmental monitoring [4], surveillance, underwater localization and many more [5]. To perform high-level multi-

The associate editor coordinating the review of this manuscript and approving it for publication was Bo Zhang.

robot tasks, it is vital to maintain high-quality of information exchange among the mobile robots [4]. The data collection efficiency is important since the wireless mobile robot swarms are more vulnerable to be failed due to their unstable network connection and limited resources [6]. Especially, in disaster scenarios where the data loss may occur frequently, it is a big challenge to guarantee reliability of real-time data collection.

With the advent of network coding, various coding strategies have been designed to enhance the data survivability in the wireless multiple access network. Among them, the Growth codes increases the persistence of data through a dynamically changing codeword degree distribution, and ensures the data collection efficiency.

In the Growth codes, time is divided into rounds. As the authors emphasized, this division is merely to facilitate the description and evaluation of techniques. Consider a sink that
is attempting to collect data over a series of rounds. The symbols can be XOR’d together to generate fixed-size codewords. The number of these symbols is referred to as the degree of the codeword. When few data have been collected by the sink, it is better to receive degree-1 codewords for immediate decoding. As the amount of data increases, the degree of codewords progressively increases for the better decoding probability. The set of symbols that form the codeword are selected uniformly at random. And at each round, the node exchanges a codeword with a neighbor randomly.

However, the totally random strategy in the Growth codes protocol results in a large number of redundant codewords in the network in the later period of data collection. These redundant codewords only contain symbols that had already recovered in the sink. The sink can collection one codeword in each round. As a result, if the sink receives a redundant codeword, this collection round is wasted. Therefore, as the number of recovered symbols increases, the probability that the sink receives these duplicate symbol copies will continue to increase. And the codewords distributed around the sink have a direct effect on data collection efficiency.

Inspired by these phenomena, if the proportion of redundant codewords around the sink can be reduced, the probability that sink can recover new symbols will increase accordingly, which improves the data collection efficiency. The question is that how to use limited information to judge whether a codeword is redundant.

Starting from the idea of reducing the proportion of redundant data, we propose a sectional configured data collection strategy based on the Growth codes, called the SCGC protocol. Some neighbors of sink are set as cache nodes, which can use the information from sink to filter the redundant codewords before forwarding to sink.

The structure of the paper is as follows: In Chapter II, some related works will be introduced, and the motivation of designing data collection strategy will be described; in Chapter III, we will introduce the problem that data collection efficiency declines in the later period of the Growth codes protocol, and analyze the causes. Finally a sectional configured network model will be presented; in Chapter IV, the data collection strategy based on the proposed network model called the SCGC protocol will be presented as well as its general algorithm framework; in Chapter V, for the information exchanged between the cache nodes and the sink, mathematical analysis will be used to build constraints to balance the information update overhead with its validity, an information update strategy of SCGC is designed; in Chapter VI, the simulation results of various different network scenarios will be presented to analyze the performance of the SCGC protocol; In Chapter VII, a brief summary is given.

II. RELATED WORKS

Data collection is a basic function of most wireless sensor networks. Since R. Ahlswede et al. first proposed the concept of network coding [1], it was applied to many applications. Nowadays, many scholars apply this technology to the data collection in WSN [13], [14], and proposed many data collection protocols with various characteristics and uses [21]–[23]. Since we are interested in the rapid data collection in the wireless many access network, some data collection protocols will be briefly introduced.

A. DELAYED DATA COLLECTION PROTOCOLS

Digital fountain codes, as a class of erasure-recovering codes, the original purpose is to improve the throughput of point-to-point network. In fountain codes, a potentially infinite number of coded symbols can be generated from $k$ input symbols. This coding method has been applied to many wireless sensor network scenarios [7]–[12]. Among them, LT codes [7] can recover all $k$ input symbols with a little more than $k$ output symbols. The intermediate performance of LT codes is poor when the number of recovered symbols is not sufficient for decoding.

In order to improve the intermediate performance of the LT codes, Beimel et al. [15] using feedbacks to choose a fixed degree for future output symbols. Hagedorn et al. [16] modified the robust soliton distribution of the LT codes. Hashemi et al. [17] proposed a protocol with the distance-type feedback messages to select input symbols no uniformly.

Although the above protocols can effectively improve the intermediate recovery rate of data, they are based on a large number of feedbacks and stable feedback channels. Also, they assume that input symbols are located at a centralized point. In the distributed networks, especially in the disaster scenarios, it is impractical to aggregate all or a substantial subset of the data at one point. These strategies do not perform as expected.

B. RAPID DATA COLLECTION PROTOCOLS

In the emergency or disaster scenarios, data should be transmitted to sink and recovered quickly because of the unpredictable failure of sensor nodes. Kamra et al. [18] systematically analyzed the data persistence in disaster scenarios, and proposed a novel data encoding and distribution technique called the Growth codes. With the Growth codes protocol, data is more likely to reach to the sink, even as sensor nodes fail.

In the Growth codes, a d-degree codeword, is immediately decodable if d-1 of these initial symbols are already recovered. This procedure can achieve a high intermediate recovery rate of data. Since the Growth codes can improve the intermediate performance, Thomos et al. [19] applied it to maximize video transmission over error resilient channel.

If all d initial symbols have been recovered, this codeword is redundant and no use for future data recovery. As the number of recovered symbols increases, the degree of codeword also needs to be increased in order to maintain the high decodable probability. In the later period of data collection, there will be a large amount of redundant data in the network [20], which will affect the efficiency of data collection.

Zhang et al. [20] proposed the Random feedback digestion (RFDG) model, which uses feedbacks to eliminate...
TABLE 1. The conclusion of related works.

| SCHEME          | CHARACTERISTICS                                                                 | FEEDBACK | LIMITATIONS                                                                 |
|-----------------|---------------------------------------------------------------------------------|----------|-----------------------------------------------------------------------------|
| LT codes        | Symbols are XOR of based robust soliton degree distribution to provide reliable communication without ARQ. | NO       | A sufficient number of codewords are needed for decoding, and intermediate performance of data collection is poor. |
| RT codes and SLT codes | The number of recovered symbols is send back to the transmitter to modify the degree distribution at the encoder. | YES      | Based on stable feedback channels and lacks a general method that enables the decoder to tune the feedback transmission rate. |
| Growth codes    | The degree of codeword increases as the number of decoded symbols increases to ensure the high instantaneous decoding probability of each codeword. | NO       | The random coding strategy generates many invalid codewords, and the existence of these codewords as well as its impact on data collection efficiency have not be considered. |

redundant codewords. In order to control the feedback messages, three different feedback mechanisms are proposed, and the degree distribution is also modified, which improves the data collection efficiency. However, as the authors stated in the article, there will be a large amount of additional feedback messages in the later period of collection, which causes a certain burden on the network.

C. SUMMARY

The process of data collection in wireless network becomes complex since there are multiple access. In the disaster scenarios where the data needs to be collected and recovered as soon as possible, the data intermediate recovery rate is an important criterion to measure a data collection scheme.

Table 1 shows a brief summary of some related works [7], [15], [16], [18], including their characteristics and limitations.

The traditional LT codes needs to collect slightly more than the original number of symbols to successfully recover all the data, and it is not suitable for this real-time collected scenario. The protocols proposed to improve the intermediate performance of the LT codes [15], [16] need to utilize stable feedback channels, which is difficult to be guaranteed in disaster scenarios. The Growth codes protocol with good data collection efficiency performs poorly in sparse networks [20]. Moreover, in a disaster scenario, excessive design of topology or a centralized control of the network can result in too much network overhead.

Therefore, for the purpose of ensuring data persistence and high collection efficiency without excessive manipulation of the wireless multiple access network, we propose a sectional configured data collection strategy based on the Growth codes, called the SCGC protocol. The protocol mainly includes the following parts:

1) Sectional configured network model. Setting the neighbor nodes of the sink as cache nodes to guarantee the reliability of information exchanging. This model only makes minor modifications to the zero-configuration network, the topology overhead can be effectively controlled.

2) Codeword component judgment mechanism. The purpose of setting cache nodes around the sink is to use these nodes to filter the codewords that will be forwarded to the sink later. By deleting these redundant codewords that waste the collection rounds, the data collection efficiency can be improved. The codeword component judgment mechanism is designed for cache nodes to judge out and delete the redundant codewords.

3) Cache nodes scheduling mechanism. Due to the existence of multiple cache nodes, forwarding conflicts are bound to occur. Using the scheduling mechanism, the sink can schedule the forwarding order of the cache nodes according to the known information about cache nodes. In this way, conflicts can be well avoided and energy consumption can be reduced. Also, the codewords that the sink will receive can be scheduled, and the probability that the received codewords can be decoded immediately can be improved, thereby achieving the purpose of improving the intermediate recovery rate of data.

4) Information update strategy. In order to complete the above two mechanisms, the information between the cache nodes and the sink needs to be synchronized. The timelier the information is updated, the better the effect of filtering out the redundant codewords, and the higher the intermediate recovery rate of data. However, each synchronization consumes the rounds originally used to collect the codewords which will decline the intermediate performance. Therefore, the information update strategy is designed to balance the information update overhead with the codeword filtering effect in the cache nodes. As a result, high accuracy of the information is ensured while minimizing the impact of updating on the data collection efficiency.

III. PROBLEM DESCRIPTION AND THE DESIGN OF SECTIONAL CONFIGURED NETWORK MODEL

In this chapter, the possible reason of data collection efficiency declining is first analyzed. On this basis, the motivation and overall idea of designing the SCGC protocol are expounded, that is, why to set the cache nodes around the sink, and how these nodes can use the information about recovered symbols recorded by the sink to ensure the data collection efficiency during the whole period. Finally, a sectional configured network model is given.

A. DESCRIPTION AND ANALYSIS OF THE PROBLEM

In order to standardize the decoding procedure in the Growth codes, a terminology of codeword-distance [18] needs to be first introduced:
Lemma 1: Given a set $X$ of original symbols and a codeword symbol $s$ of degree $d$, the distance of $s$ from set $X$, $\text{dist}(s, X)$, is the number of symbols out of the $d$ symbols XOR’d together to form $s$, which are not present in $X$.

Based on this terminology, the procedure of decoder can be simply described as:

Initially the set $X$ of recovered symbols is empty. For a received codeword $s$, if $\text{dist}(s, X) = 0$, the decoder does nothing but just throw away it as a redundant codeword; if $\text{dist}(s, X) = 1$, the decoder can decode a new symbol and add it to set $X$; if $\text{dist}(s, X) > 1$, the decoder keeps this codeword in a queue waiting for later decode. This procedure repeats in each round until all original symbols have been recovered.

The sink is attempting to collect data over a series of rounds. In each round, the sink receives one codeword without knowing whether it is redundant or not. And a redundant codeword will waste one round, which affects the data intermediate recovery rate.

Fig. 1 depicts the proportion of different types of these codewords (codewords other than redundant codewords are called valid codewords) changing with collection round. The abscissa represents the number of collected codewords in the Growth code protocol.

Clearly, at the later period, as more and more symbols have been recovered, the probability of receiving a valid codeword is decreasing. This situation is similar to the Coupon Collector’s effect, which leads to an unavoidable decline of data recovery efficiency in the Growth codes.

According to the correspondence between the proportion of redundant codewords received by the sink and the data recovery efficiency, we hope to alleviate the decline in data recovery efficiency by filtering redundant codewords. The question is how to pre-judge the redundant codewords out and reduce the possibility of forwarding them to sink. We proposed a data collection strategy based on the sectional configured network. The neighbor nodes of the sink are set as cache nodes. The sink will feedback the information about recovered symbols to the cache nodes. Based on these information, the cache nodes can judge out and delete the redundant codewords. The possibility for sink to receive a redundant codeword will be decreased, improving the data recovery efficiency.

IV. THE SCGC PROTOCOL

In this chapter, we will introduce a data collection strategy based on the sectional configured network, the SCGC protocol, including codeword component judgment mechanism and cache nodes scheduling mechanism. We will describe in detail how the cache nodes and the sink use these mechanisms.
to judge and delete redundant codewords, reduce forwarding conflicts, and improve data intermediate recovery rate.

A. CODEWORD COMPONENT JUDGMENT MECHANISM
To filter out the redundant codewords, the cache nodes first need to analyze the codeword component and judge the codeword-type of any monitored codeword. The classification of any d-degree codeword s based on codeword-distance is as follows:

1) If \( \text{dist}(s, X) = 0 \), it means all the d initial symbols constituting the codeword s have been recovered at sink. This codeword has no help in recovering a new symbol and should be filtered. It is a redundant codeword.

2) If \( \text{dist}(s, X) = 1 \), it means only one of the d initial symbols has not been recovered. As a result, sink can decode a new symbol from this codeword as soon as receiving it. It is a decodable codeword.

3) If \( \text{dist}(s, X) > 1 \), it means more than one initial symbol in d symbols has not been recovered. This codeword cannot be immediately decoded, and the sink will add it in a queue waiting for later decode. It is a bench codeword.

4) In addition, since the decodable codeword and the bench codeword are helpful in decoding new symbols, both can also defined as valid codeword collectively.

The cache nodes use the following strategy to filter a monitored codeword.

First, the sink synchronizes the recovered symbols set \( X \) through a filter-bitmap to the cache nodes. Based on this information, each cache node can calculate the codeword-distance. They filter and throw out the redundant codewords, and reserve the valid codewords. At the time of forwarding, the cache node forwards a reserved codeword such that the codeword-distance is the minimum. Once the filter-bitmap is updated, the cache node recalculates the codeword-distance of the reserved codewords and filters redundant codewords.

This behavior of all cache nodes can be abstracted into a \( \text{filter} \ F \): For a codeword \( s \), if \( s \) is a redundant codeword, the \( \text{filter} \ F \) will discard it; if \( s \) is a valid codeword, the \( \text{filter} \ F \) will add it into forwarding queue. The pseudo code of the codeword component judgment algorithm is shown in Algorithm 1.

B. CACHE NODES SCHEDULING MECHANISM
There are usually more than one cache nodes around the sink, and a collision will happen. And the data intermediate recovery rate has a great relationship with the received codeword-type. If sink receives the decodable codeword preferentially and recovers a new symbol immediately, the data collection efficiency will close to the theoretical upper bound. For this purpose, the cache nodes scheduling mechanism is designed to control the forwarding order of all the cache nodes.

1) THE UDSP AND TDSP FORWARDING MECHANISM
The forwarding mechanism of cache nodes (as shown in Fig. 3) is mainly divided into two periods: the update data to sink period (UDSP) and the transfer data to sink period (TDSP). These two periods interchange periodically and are synchronized by the sink. The detail of these two periods will be described to explain how to use this forwarding mechanism to schedule cache nodes to ensure a high data intermediate recovery rate.

Algorithm 1 Codeword Component Judgment Mechanism

**Input:**
The latest filter-bitmap packet from sink.
The reserved codewords.

**Output:**
The latest set of reserved codewords.

1: The sink sends the latest filter-bitmap packet to cache nodes.
2: for each cache node \( k \) do
3: receive the filter-bitmap packet from sink,
4: update the previous filter-bitmap with the latest set \( X \) in packet.
5: calculate codeword-distance \( \text{dist}(s_i, X) \) of each reserved codeword \( s_i \) with the latest filter-bitmap.
6: if \( \text{dist}(s_i, X) = 0 \) then
7: throws \( s_i \)
8: else
9: reserves \( s_i \)
10: end if
11: end for
12: return The latest set of reserved codewords.

FIGURE 3. The forwarding mechanism.
Algorithm 2: The Update Data to Sink Period (UDSP)

**Input:**
- The latest filter-bitmap packet from sink.
- The reserved codewords.

**Output:**
- The latest forwarding order list \( L \).

1: The sink sends the latest filter-bitmap packet to cache nodes.
2: for each cache node \( k \) do
3: receive the filter-bitmap packet from sink,
4: update the previous filter-bitmap with the latest set \( X \) in packet,
5: clear previous codeword-distance set \( D_k \),
6: calculate codeword-distance \( \text{dist}(s_i, X) \) of each reserved codeword \( s_i \) with the latest filter-bitmap.
7: if \( \text{dist}(s_i, X) = 0 \) then
8: throws \( s_i \)
9: else
10: reserves \( s_i \)
11: end if
12: add \( \text{dist}(s_i, X) \) to codeword-distance set \( D_k \)
13: send the packet which contains the latest set \( D_k \)
14: end for
15: The sink constructs the forwarding order list \( L \) by all the set \( D \) from cache nodes and broadcasts it back.

Algorithm 3: The Transfer Data to Sink Period (TDSP)

**Input:**
- The latest forwarding order list \( L \).

**Output:**
- The latest forwarding order packet.

1: The sink sends the forwarding order packet to cache nodes.
2: for each cache node \( k \) do
3: receive the forwarding order packet from sink,
4: update the previous forwarding order list \( L \),
5: check its scheduled round in list \( L \).
6: if node \( k \) is scheduled in the current round then
7: forward the codeword has the minimum codeword-distance
8: else
9: monitor the exchange of codewords between sensor nodes
10: end if
11: end for
12: The sink decodes codewords and adds new recovered symbol to set \( X \),
13: puts the set \( X \) in the latest filter-bitmap packet and broadcasts it.

**b: THE TRANSFER DATA TO SINK PERIOD (TDSP)**

In TDSP, cache nodes find their transmission round in the forwarding order list \( L \) and only transmit during their scheduled time. Then, in each round, only one codeword will be forwarded to the sink, preventing the collisions. And this codeword is probably the optimal codeword that can recover a new symbol, which will effectively improve the utilization of the output channel and achieve the purpose of ensuring a high data intermediate recovery rate.

Once a new symbol is decoded, the sink will add it to set \( X \). At the end of TDSP, the sink broadcasts the latest filter-bitmap packet to cache nodes, and period changes to UDSP. The pseudo code of TDSP is shown in Algorithm 3.

**2) THE INFORMATION UPDATE STRATEGY OF SCGC**

Noted worthy, the cache nodes always monitor codewords exchanging between the neighboring sensor nodes, and save valid codewords. The sink decodes decodable codewords or saves bench codewords after receiving it. As time goes on, information will be out of sync between the cache nodes and the sink, causing the deviation of codeword-distance calculated by cache nodes and sink. It is very likely that the decodable codeword forwarded by the cache node is in fact a redundant codeword in the sink, affecting the actual data collection efficiency.

To control these deviations, the information should be updated. Ideally, in order to ensure that the codeword-distance calculated by the cache node is not deviated from the sink, once the sink recovers a new symbol, the latest recovered symbols set \( X \) should be updated to the cache nodes, and
TDSP switches to UDSP. However, in UDSP, the cache node does not forward the codeword to sink, so frequent period switching will cause the codeword accumulation in the cache nodes. More seriously, the sink has to receive codeword-distance packets which do no help to decode symbols but just waste the collect round. The performance of the SCGC protocol will decrease sharply. The information update strategy is designed to balance the information update overhead with the codeword filtering effect in the cache nodes.

Although a dynamic update cycle is the more scientific for information updating, it is not the main focus in this work. We choose a fixed update cycle length $T$, in other words, two periods interchange periodically once a fixed $T$ new symbols is recovered.

To ensure the reasonableness of the pre-selected update cycle, some constraints will be constructed to select appropriate values for different network conditions. This part of the work will be completed in the next chapter.

C. PACKET MODEL

In the SCGC protocol, information needs to be exchanged between the cache nodes and the sink, including the recovered symbol set $X$, the codeword-distance set $D$ and the forwarding order list $L$. Therefore, besides the ordinary codeword packets, it is necessary to forward other type packets. In this section, the packet models will be described.

1) THE NORMAL CODEWORD PACKET

Fig. 4 shows the structure of a normal codeword packet. This packet is exchanged between sensor nodes and forwarded by cache nodes to sink in TDSP. The first bit of the first byte is 0 indicates that this is a normal codeword packet. The next $N$ bits indicates the initial symbols that consist this codeword: a 1 bit signifies the presence of a particular component and 0 specifies the absence.

2) THE FILTER-BITMAP PACKET

Fig. 5 shows the structure of a filter-bitmap packet containing the recovered symbols set $X$. The first bit of the packet is 1 to indicate that this is a special packet. An $N - bits$ bitmap is used to indicate the recovered symbols set $X$: a 1 bit signifies this symbol is recovered and 0 specifies this symbol is unrecovered. The codeword-distance can be calculated by these $N$ bits.

3) THE CODEWORD-DISTANCE PACKET

Fig. 6 shows the structure of a codeword-distance packet containing the codeword-distance set $D$. Similarly, the first bit of the packet is 1, indicating that this is a special packet. The distance of each codeword needs to be represented by $\log(N)$ bits. Since the distance is the number of symbols out of the $d$ symbols XOR’d together to form the codeword, it will not consumed too much space to list the codeword-distance.

4) THE FORWARDING-ORDER LIST PACKET

Fig. 7 shows the structure of a forwarding-order list packet sent by the sink to the cache nodes. Similarly, the first bit of the packet is 1, indicating that this is a special packet. The cache node IDs each of size $\log(N)$ are packed into as few bytes as possible. The length of the forwarding order list $L$ is related to the length of update cycle.

Since the cache nodes need to receive and store different kinds of packets, the storage overhead will be higher than other sensor nodes, but these overhead can be ignored among the whole network.

V. THE INFORMATION UPDATE STRATEGY OF SCGC

The interchange time of two period has an impact on data collection efficiency. This effect is mainly due to the fact that the recovered symbol set $X$ saved by the cache node cannot be synchronized with the sink timely. Based on this unsynchronized set, the codeword-distance calculated by the cache nodes may have deviation.

The more timely the information is updated, the better the effect of filtering out the redundant codewords, and the higher the intermediate recovery rate of data. However, each synchronization consumes the rounds originally used to collect the codewords which will decline the intermediate performance.

In this chapter, through the mathematical analysis of the filtering ability of the filter $F$ and the energy consumption of the cache nodes, an information update strategy is designed to balance the information update overhead with the codeword filtering effect in the cache nodes. As a result, high accuracy...
of the information is ensured while minimizing the impact of updating on the data collection efficiency. The constructed constraints can be used to select appropriate values of update cycle for different network conditions.

A. THE DESIGN OF UPDATE STRATEGY

As mentioned previously, the behavior of all cache nodes can be abstracted into a filter $F$. For a codeword $s$, if $s$ is a redundant codeword, the filter $F$ will discard it; if $s$ is a valid codeword, the filter $F$ will add it into forwarding queue. At each slot in TDSP, the filter $F$ will forward a codeword which has a minimum codeword-distance.

To reduce the deviation of codeword-distance, the recovered symbols set $X$ should be updated to cache nodes to re-calculate the codeword-distance. Next, the filtering effect of the filter $F$ will be quantitatively analyzed to select the update cycle more reasonably for different network conditions.

The analysis is based on the degree distribution in the Growth codes, some conclusions [24] need to be introduced here.

**Theorem 1:** Let $\rho_{r,d}$ be the probability of successfully decoding a degree $d$ symbol when $r$ symbols have already been recovered. Then,

$$\rho_{r,d} = \frac{\binom{d-1}{r}}{\binom{N}{d}}. \tag{1}$$

**Theorem 2:** Let $R_i$ represent the number of symbols recovered by a sink when codewords of size greater than $i$ provide a greater likelihood for providing recovery than those of degree less than $i$.

$$R_i = \frac{N - i}{N - 1}, \cdots, R_i = \frac{iN - 1}{i + 1}, \quad \forall i \in [1, N - 1] \tag{2}$$

Based on the above theorems, the following conclusions address:

**Lemma 2:** Let $\rho_{r,d}$ be the probability of a $d$-degree codeword discarded by the filter $F$ when $r$ symbols have already been recovered. Then,

$$\rho_{r,d} = \frac{\binom{d}{r}}{\binom{N}{d}}. \tag{3}$$

**Proof:** The $d$ symbols of the $d$-degree codeword are assumed to be distinct and uniformly chosen without replacement from the $N$ symbols. A codeword will be discarded if all the $d$ components of the codeword have already been recovered. The number of ways of choosing a $d$ degree symbol such that the component symbols are distinct and are spread uniformly randomly is $\binom{N}{d}$. For a $d$-degree codeword, the number of ways of choosing $d$ components from the set of $r$ recovered symbols is $\binom{r}{d}$. The probability that for a $d$-degree codeword, all $d$ components are from the set of $r$ recovered symbols is $\rho_{r,d} = \frac{\binom{r}{d}}{\binom{N}{d}}$. ■

**Lemma 3:** Let $p_{kT,d}$ represent the probability of a $d$-degree codeword can be filtered by the filter $F$ before the $k^{th}$ update of the recovered symbols set $X$. Then,

$$p_{kT,d} = \frac{(k-1)^{T-d}}{\binom{N}{d}}. \tag{4}$$

**Proof:** Once $T$ symbols have been recovered in sink, the recovered symbols set $X$ will updated to cache nodes. Before the $k^{th}$ update, the size of set $X$ recorded in filter $F$ is $(k-1)T$. According to lemma 2, the probability of a $d$-degree codeword can be filtered by the filter $F$ is $p_{kT,d} = \frac{(k-1)^{T-d}}{\binom{N}{d}}$. ■

**Lemma 4:** For the same $d$-degree codeword, $p_{kT,d} > p_{jT,d}$ for $i > j$.

**Proof:** Since $i > j$, suppose $i = j + 1$, then:

$$p_{kT,d} \div p_{jT,d} = \frac{(j+1)^T}{\binom{N}{d}} \div \frac{(jT)^d}{\binom{N}{d}} = \frac{(j+1)^T}{(jT)^d} = \frac{(j+1)^T}{(j+1)^T} \times d!(jT - d)! \times \frac{(jT)!}{(jT + kT)(jT + kT - 1) \cdots (jT + 1)} = \frac{(jT + kT - d)(jT + kT - 1 - d) \cdots (jT + 1 - d)}{1} > 1$$

From lemma 4, we can conclude that the more times the set $X$ is updated, the more likely a codeword is filtered.

**Lemma 5:** When the number of symbols recovered by a sink is $R_1 = \frac{N - 1}{2}, \cdots, R_T = \frac{iN - 1}{i + 1}, \forall i \in [1, N - 1]$.

Based on the above, the following conclusions address:

**Lemma 6:** Let $S$ represent the number of symbols discarded by the filter $F$ when $r$ symbols have already been recovered. Then,

$$S = \sum_{i=1}^{T} \sum_{t=(i-1)T}^{iT-1} N \times \frac{(j-1)^{T-d}}{\binom{N}{d}}. \tag{5}$$

**Proof:** 1. For recovering 0 to $T$ symbols, the filter-bitmap in cache nodes has not been updated. The size of recovered symbols set $X$ in filter $F$ is zero which means that filter $F$ has no filtering capability. In this case, in order to recover T symbols while all the codewords have to be degree 1, it is equivalent to Coupon Collector’s Problem. To get the first T distinct coupons, one needs to collect $\sum_{i=0}^{T-1} \frac{N}{N-i}$ coupons in expectation.

2. The expected number of codewords required is $\sum_{i=0}^{T-1} \frac{N}{N-i}$.

Hence the expected number of codewords required is

$$\sum_{i=0}^{T-1} \frac{N}{N-i}.$$

For recovering $T + 1$ to $2T$ symbols, the filter-bitmap in cache nodes is updated once. If there is no filter $F$, one needs to collect $\sum_{i=T}^{2T-1} \frac{N}{N-i} = \sum_{i=T}^{2T-1} \frac{N}{N-i}$ to recover new T symbols. Since the filter $F$ knows that T symbols have been recovered, according to (3) in lemma 2, the probability of codewords will be filtered is $p_{2T,d} = \frac{\binom{2T-1}{d}}{\binom{N}{d}}$. ■

$\sum_{i=T}^{2T-1} \frac{N}{N-i} \times p_{2T,d} = \sum_{i=T}^{2T-1} \frac{N}{N-i} \times \frac{\binom{2T-1}{d}}{\binom{N}{d}}$
3. For recovering \((k - 1)T + 1\) to \(kT\) symbols, the filter-bitmap in cache nodes is updated \((k - 1)\) times. The size of the recovered symbols set \(X\) in the filter \(F\) is \((k - 1)T\). Then the probability of codewords will be filtered is \(p_{kT, d} = \frac{(k-1)T}{\binom{N}{d}}\), the number of filtered codewords is:

\[
\sum_{i=(k-1)T}^{kT-1} \frac{N}{N-i} \times p_{kT, d} = \sum_{i=(k-1)T}^{kT-1} \frac{N}{N-i} \times \frac{(k-1)T}{\binom{N}{d}}
\]

In summary, to recover \(R_1 = \frac{N}{N-1}\) symbols, \(|R_1/T| = k\), the set \(X\) has been updated \(k\)-1 times. Then the expected number of codewords that can be filtered by filter \(F\) is \(S_1 = \sum_{j=1}^{k} \sum_{i=(j-1)T}^{jT-1} \frac{N}{N-i} \times \frac{(j-1)T}{\binom{N}{d}}\).

Let

\[
A_j = \sum_{i=(j-1)T}^{jT-1} \frac{N}{N-i} \times \frac{(j-1)T}{\binom{N}{d}} \quad (6)
\]

**Lemma 6:** To recover all \(N\) symbols, \(|N/T| = \delta\), the expectation number of codewords that filtered by filter \(F\) is:

\[
S = S_1 + \sum_{j=1}^{\delta} A_j \quad (7)
\]

where \(S_1 = \sum_{j=1}^{\delta} \sum_{i=(j-1)T}^{jT-1} \frac{N}{N-i} \times \frac{(j-1)T}{\binom{N}{d}}\).

**Proof:** After recovering \(R_1 = \frac{N}{N-1}\) symbols, according to degree distribution, the degree \(d\) of codewords is greater than 1. When \(r\) symbols have been recovered, the next degree codeword will have distance 1 from the set of recovered symbols with probability \(\frac{C_r}{C_d} (N-i) \binom{N-i}{d} \). To recover the next symbol, the expected number of codewords required is \(\frac{C_r}{C_d} (N-i) \binom{N-i}{d} \).

For recovering \(kT+1\) to \((k+1)T\) symbols, the size of the recovered symbols set \(X\) in the filter \(F\) is \(kT\). Then the filtered probability of a d-degree codeword is \(p_{(k+1)T, d} = \frac{\binom{N}{d}}{\binom{N}{d}}\), the number of filtered codewords is:

\[
\sum_{i=KT}^{(k+1)T-1} \frac{\binom{N}{d}}{\binom{N-i}{d}} \times p_{(k+1)T, d} = \sum_{i=KT}^{(k+1)T-1} \frac{\binom{N}{d}}{\binom{N-i}{d}} \times \frac{kT}{\binom{N}{d}} = A_k
\]

In summary, to recover all \(N\) symbols, \(|N/T| = \delta\), the expected number of codewords that the filter \(F\) can filter is

\[
S = S_1 + \sum_{j=1}^{\delta} A_j \quad \blacksquare
\]

**Lemma 7:** Let \(\rho_{r, d}\) be the probability of successfully decoding a new symbol from received d-degree codeword when \(r\) symbols have already been recovered. Then \(\delta_{r, d} = 1 - (1 - \rho_{r, d})^\delta\), where \(\theta\) is the total number of codewords stored in all cache nodes.

**Proof:** According to theorem 5.1, the probability of successfully decoding a d-degree codeword when \(r\) symbols have already been recovered is \(\rho_{r, d}\). At each round in TDSP, the filter \(F\) forwards a codeword in all \(\mathcal{E}\) codewords. To recover a new symbol, it is necessary to ensure that at least one codeword is decodable. For a d-degree codeword, the probability that it cannot be decoded immediately is \((1 - \rho_{r, d})\). The probability that all \(\theta\) codewords cannot be decoded immediately is \((1 - \rho_{r, d})^\delta\). Therefore, the probability of successfully decoding a new symbol from received d-degree codeword is \(\delta_{r, d} = 1 - (1 - \rho_{r, d})^\delta\).

According to lemma 7, the more the number of cache nodes, the more the number of codewords can be filtered. Also, the more cache nodes exist, the higher the probability of recovering new symbols. At the same time, the information is updated in UDSP, where cache nodes will not forward codewords to sink. If the information has updated \(\delta\) times, for \(\lambda\) cache nodes, it will at least cost a total of \(\delta \lambda\) rounds. These rounds are called as the update overhead. Therefore, if the number of cache nodes is too large, it will affect the receiving of codewords and also affect the data collection efficiency.

According to the above analysis, it is necessary to reduce the update overhead while ensuring the filtering effect of the filter \(F\). Based on this, the first constraint is that the total number of filtered codewords should larger than update overhead. This constraint can be described as:

\[
S - \delta \lambda > 0 \quad (8)
\]

**B. THE ANALYSIS OF NODE ENERGY CONSUMPTION**

A constraint of update cycle based on the balance of the filtering effect and the update overhead has been proposed. Besides, the behavior of cache nodes is different to sensor nodes, resulting difference in energy consumption. Since the SCGC protocol is based on cache nodes, if the cache nodes are failed prematurely, it will affect the data collection efficiency. As a result, the node energy consumption will be analyzed to construct another constraint.

1) **ENERGY CONSUMPTION MODEL**

Since the consumption of executing computer instructions is far less than transmission packets, for convenience, we only consider the energy consumption of the node to send and receive packets. We use the radio model in [25] to calculate the energy consumption. The energy consumed by node to send and receive \(n\) bits of data can be expressed as:

\[
E_{\text{send}} = nE_{\text{circuit}} + nE_{\text{amp}}d^\alpha
\]

\[
E_{\text{receive}} = nE_{\text{circuit}}
\]

Assumed that the transmission distance of a packet is equal to the node transmission radius \(r\), a certain range within which nodes can communicate. Since this range is not large, set \(\alpha = 2\), and the amplification factor of the signal amplifier is \(k_{\text{amp}} = 10\) pJ/bit/m².

Table 2 shows the experimental parameters.
Table 2. Experimental parameters.

| PARAMETER                  | VALUE     |
|----------------------------|-----------|
| Size of packet head        | 128 bit   |
| Size of sensor data unit   | 32 bit    |
| Size of filter-bitmap      | 500 bit   |
| \(E_{\text{circuit}}\)    | 50 nJ/bit |
| \(e_{\text{amf}}\)        | 10 pJ/bit/m² |
| \(e_{\text{amp}}\)        | 0.0013 pJ/bit/m³ |

Based on these parameters, the energy consumptions of sending and receiving various packets can be calculated.

1. The size of a codeword packet is 20 bytes. Each sensor node exchanges codeword packet at most once per round. The corresponding energy consumption is:

\[
E_{\text{send}} = 160 \times 50n j + 160 \times 10 \times 10^{-3} n J / m^2 \times r^2
\]  
(11)

\[
E_{\text{receive}} = 640 \times 50n j
\]  
(12)

2. In the network with 500 nodes, the size of the filter-bitmap packet is about 80 bytes. Each time the cache node needs to update information with sink, it receives one filter-bitmap packet. The corresponding energy consumption \(E_{\text{filter}}\) is:

\[
E_{\text{filter}} = 640 \times 50n j
\]  
(13)

3. The size of codeword-distance list packet is not fixed, which is related to the total number of codewords stored in cache nodes. The size of codeword-distance list \(L\) is \(l_{\text{dist}}\), and suppose each cache node can at most store \(C_0\) codewords. So, \(0 \leq l_{\text{dist}} \leq C_0 \times 8\text{ bit}\). Each time the cache node needs to update information with sink, it sends one codeword-distance list packet. The corresponding energy consumption \(E_{\text{dist}}\) is:

\[
E_{\text{dist}} = (l_{\text{dist}} + 128) \times 50n j
\]  
\[
+(l_{\text{dist}} + 128) \times 10 \times 10^{-3} n J / m^2 \times r^2
\]  
(14)

4. The size of forwarding-order list packet is \(l_{\text{list}}\), just like \(l_{\text{dist}}\), it is not fixed but does not exceed the total number of stored codewords in cache nodes, so, \(0 \leq l_{\text{list}} \leq \lambda C_0 \times 8\text{ bit}\). Each time the cache node needs to update information with sink, it receives one forwarding-order list packet. The corresponding energy consumption \(E_{\text{list}}\) is:

\[
E_{\text{list}} = (l_{\text{list}} + 128) \times 50n j
\]  
(15)

2) NORMAL SENSOR NODES

Assuming that the total round of data collection is \(R_{\text{sum}}\), the expected energy consumption of each sensor node can be expressed as:

\[
E_{\text{sensor}} = (E_{\text{send}} + E_{\text{receive}}) \times R_{\text{sum}}
\]  
(16)

Therefore, under the condition that the packet size is fixed, and other effects are ignored, the energy consumption of the sensor node is only related to the time of data collection.

3) CACHE NODES

For the cache node, since each \(T\) symbols have been recovered, the information will be updated. For recovering \(N\) symbols, \([N/T]\) = \(\vartheta\) times updates have been performed. Assuming that the total rounds of data collection is \(R_{\text{sum}}\) and the number of cache nodes is \(\lambda\). The length of UDSP is \(\vartheta \lambda\) and the length of TDSP is \((R_{\text{sum}} - \vartheta \lambda) / \lambda\).

The number of various packets that each cache node needs to send and receive is as follows:

1. Receives \(\vartheta\) filter-bitmap packets and \(\vartheta\) forwarding-order list packets;
2. Sends \(\vartheta\) codeword-distance packets and average \((R_{\text{sum}} - \vartheta \lambda) / \lambda\) codeword packets;
3. Since the cache nodes maintain monitoring the exchange of codewords between sensor nodes, \(R_{\text{sum}}\) codeword packets are monitored at most.

In summary, the expectation of energy consumption per cache node can be expressed as:

\[
E_{\text{cache}} = E_{\text{re}} + E_{\text{se}}
\]
\[
= (E_{\text{filter}} \times \vartheta + E_{\text{list}} \times \vartheta + E_{\text{receive}} \times R_{\text{sum}})
\]
\[
+(E_{\text{dist}} \times \vartheta + E_{\text{send}} \times (R_{\text{sum}} - \vartheta \lambda) / \lambda)
\]  
(17)

The \(E_{\text{cache}}\) is related to the times of update and the number of cache nodes. The more frequent the update, the greater the \(E_{\text{cache}}\). For the stability of the network, it is necessary to ensure that \(E_{\text{cache}} \leq E_{\text{sensor}}\), which is the other constraint in our scheme. To sum up, based on the filtering effect and the energy consumption, two constraints can be stated as follows:

\[
S - \vartheta \lambda > 0
\]  
\[s.t. E_{\text{cache}} \leq E_{\text{sensor}}\]  
(18)

Using these constraints, the update cycle length and the number of cache nodes can be set based on various network conditions.

VI. EXPERIMENTS AND ANALYSIS

A. ANALYSIS AND SETS OF SIMULATION PARAMETERS

The basic conditions of network are equal to Growth codes. \(N = 500\) sensor nodes are randomly distributed in a \(100 \times 100\) network, and each node can store \(C_0 = 10\) codewords.

According to the previous analysis of the energy consumption and the filter ability of the cache nodes, the number of cache nodes and the length of the update cycle have a great impact on the final data collection efficiency. These two parameters need to be analyzed and set according to the network conditions.

1) ANALYSIS OF SIMULATION PARAMETERS

a: THE NUMBER OF CACHE NODES

For the selection of number \(\lambda\) of cache nodes, since the starting point is to perform a “slight” configuration on zero-configuration network. Too many cache nodes will make a lot changes to the normal model, which is unfair...
to the Growth codes protocol. Moreover, in the previous theoretical analysis, the more the number of cache nodes, the better the filtering effect, but correspondingly, the more additional consumption for information updating. Fig. 8 depicts the data collection efficiency when the cache nodes number $\lambda = 5/10/15$.

As the number of cache nodes in the network increases, the number of collection rounds to recover all symbols increases. Although, as mentioned in lemma 7, the more the number of cache nodes, the higher the probability that the sink can recover new symbols. However, since the update-overhead is associated with $\lambda$, this additional overhead increases sharply as the number of cache nodes increases, which leads to a significant decrease in data collection efficiency.

Based on the above reasons, setting the number of cache nodes $\lambda = 5$, the ratio to the total number of nodes is 0.01. In other word, the network is sectional configured, and the data collection efficiency is also stable.

**FIGURE 8.** The impact of various $\lambda$ on data collection efficiency ($\text{radius}=0.3$).

**b: THE LENGTH OF THE UPDATE CYCLE**

The update cycle length $T$ means that the sink updates the recovered symbols set $X$ to the cache nodes once $T$ new symbols are recovered.

In Fig. 8, if the selected $T$ is small, which means the update is frequent, the data collection effect will be weakened. Moreover, when the $T$ is smaller than the number of cache nodes, the total data collection may fail.

Fig. 9 depicts the data collection efficiency when update cycle length $T$ is set from 20 to 50. The simulations are performed in the dense network ($R = 0.3$) and the sparse network ($R = 0.2$).

The experimental results are consistent with the theoretical analysis: too short $T$ will affect the collection efficiency while too long $T$ can weaken the filtering effect of the cache nodes.

In addition, in Fig. 9, it seems that the data collection efficiency does not fluctuate much when the update cycle length $T$ is from 20 to 50. As defined previously, besides the redundant codeword, the decodable codeword and the bench codeword are belong to the valid codeword. Since the bench codeword which has codeword-distance $> 1$ cannot be immediately decoded, the intermediate performance is more relevant with decodable codeword. Therefore, we further calculate the proportions of different types of codewords received by sink, as depicted in Fig. 10. In Fig. 10(a), $T = 20$, while in Fig. 10(b), $T = 50$. Fig. 10(c) depicts the codewords proportions in the Growth codes protocol and it shows that the SCGC protocol can effectively filter redundant codewords.

Although the total collection rounds are similar with different $T$, however, by analyzing the proportion of different types of codewords, it can be seen that the smaller $T$ enables the cache node to provide a better filtering effect. When $T = 20$, the proportion of decodable codewords can even be guaranteed to be above 80%, and sink can recover all the symbols quickly.

Based on theoretical analysis and experimental results, in the next experiments, we set the update cycle length $T = 20$, which means that once 20 new symbols have been recovered in sink, an update will performed.

**2) THE EXPERIMENTAL PARAMETERS**

In summary, the experimental parameters in this article are given in Table 3.

According to the network setting in Growth codes, 500 sensor nodes are randomly distributed in a $100 \times 100$ network, and the storage capacity of each node is 10 codewords. The radius of $R = 0.3$ in network, i.e., the node can communicate with the nodes in the distance of 30 in a $100 \times 100$ network. After the analysis previously, we assume that the upper number of cache nodes is 5 and the length of update cycle is 20.

As stated previously, it means the sink updates the recovered...
symbols set $X$ to the cache nodes once 20 new symbols are recovered.

**B. PERFORMANCES IN VARIOUS SCENARIOS**

The data collection performance of the Growth codes protocol and the SCGC protocol will be compared in various scenarios. The main evaluation aspects are as follows:

1) The efficiency of data collection in sink.
2) The energy consumptions of cache nodes and sensor nodes.

In all simulations we use the round based JAVA simulator and unless otherwise stated. The results are an average of 1000 simulation runs.

1) THE IDEAL DISTRIBUTED NETWORK

Firstly, the performances of two protocols are compared in an ideal distributed network. In this scenario, nodes connected in a random topology and will never failed. The nodes transmission radius of $R = 0.1$, and $R = 0.3$. The efficiency is measured by the number of symbols recovered at the sink for any given number of packets received, and the results are shown in Fig. 11.

The network with a radius of $R = 0.3$ is fairly well connected, which is a dense network. Correspondingly, when $R = 0.1$, the network is a sparse network. Clearly, the SCGC protocol outperforms the Growth codes protocol both in dense and sparse network. Because the SCGC protocol does not make configuration to the whole network, in the sparse network with low node connectivity, all the data cannot be recovered in 1000 rounds.

In addition, it seems that in the early period of data collection, the SCGC protocol performs worse than the Growth codes protocol. This is mainly because at this period, the degree of codeword is 1, and the number of redundant codewords is small. In the Growth codes protocol, the sink only receives codewords and it can recover symbols more quickly. But in the SCGC protocol, in order to update information, the cache nodes need to forward some other packets to sink, leading to an updating overhead. As we mentioned earlier, this unavoidable overhead will weaken the data collection efficiency, especially in the early period. As the number of received codewords increases, the SCGC protocol can achieve higher efficiency, and sink can recover all the original symbols more quickly.

Fig. 12 depicts the proportions of different types of codewords received by sink. Compared with the Growth codes in Fig. 1, the proportion of redundant codewords can be reduced by 10-15%. What needs to be emphasized is that in the conventional scheme with the Growth codes, in order to ensure the instantaneous decoding probability of each codeword, the degree of codeword increases as the number of decoded symbols increases. There is no concept of redundant codewords which is only contains symbols that have already
recovered. In other words, to ensure such decoding probability, the existence of redundant codewords and the impact on data collection efficiency are not considered. Considering the energy efficiency and the disaster scenarios, the feedback information exists only in sink and some of its neighbor nodes in the proposed sectional configured network model. In this network model, only cache nodes can filter those redundant codewords. Therefore, the ability to filter redundant codewords is not as high as the scheme relying on a complete and reliable feedback channel. After a large number of simulations, this improvement is credible and acceptable within the limits of energy efficiency and network configuration. In other words, the redundant codewords are well filtered.

2) THE DISASTER SCENARIOS

Next, the performances of two protocols are compared in disaster scenarios. The simulations are performed in two kind of networks, and the network features are as follows:

1) The range destruction scenario: After a certain time $t$ of data collection, a disaster happens and will disable all nodes within $r$ distance of the center of the disaster;

2) The node destruction scenario: In the process of data collection, nodes will failed periodically.

The above two disaster scenarios are very common in real-world applications. More detailed descriptions of these two kinds of scenarios are stated in the corresponding experiments.

a: THE RANGE DESTRUCTION SCENARIO

In disaster scenarios, such as earthquakes, a subset of the sensor nodes may be damaged at some time. These nodes cannot participate in later data collection, and the key is how to recover the data of these nodes. The time or the impact radius of disaster will affect the data recovery efficiency.

We set the disaster occur at time $t = 250$, and the impact radius $r = 0.2$. That means all the nodes within 0.2 from the disaster center are failed at the 250 round after data collecting. Fig. 13 shows the proportion of symbols can be recovered at the sink with rounds.

It should be noted that if a cache node failed in the disaster, the sink will employ other neighbor sensor node for replacement. Since the impact occurs at the 250 round, nodes have enough time to spread data into the network, the data collection efficiency of both protocols is not severely affected. However, in total 1000 rounds, the data recovery rate at sink in the Growth codes protocol cannot reach to 100%. While in the SCGC protocol, since some redundant codewords are filtered, the Coupon Collector’s effect is well weakened. There is a high probability that all symbols will be recovered in 1000 rounds, which is beneficial for data collection on disaster scenarios.

If the disaster occurs sooner, the data of failed nodes may not have a chance to spread into the network, and these lost data will affect the data recovery rate in sink. Fig. 14 depicts the time taken to recover symbols when the disaster occurs at time $t = 50$.

We simulate the situation where the impact radius $r$ vary from 0.1 to 0.5. The number next to the data point are the amount of symbols that are eventually recovered in sink. Since the number of failed nodes is proportional to the square of the impact radius $r$, when $r = 0.5$, nearly half of the
nodes will be damaged. As a result, the final number of recovered symbols is smaller, and the time taken to recover these symbols is shorter.

Yet, comparing the different disaster impact radius, we can see that the SCGC protocol can recover the surviving data faster than the Growth codes protocol.

b: THE NODE DESTRUCTION SCENARIO

Besides the disaster scenario in which impact occurs just one time, there are also scenarios in which impact occurs continually. For example, in a monitoring network for warfare, bombs and bullets will continue to fall into the network, causing the node continually failed.

In order to simulate this scenario, we assume that starting from round 50, a random disaster occurs in each 50 rounds, and the impact radius is 0.2. As time went by, the numbers of failed nodes will increase. As shown in Fig. 14, after half of the nodes failed, the performance will decline sharply, so the disaster will stop at round 250. Fig. 15 depicts the data recovery rate in two protocols.

Fig. 15 shows that in this scenario with continuing node failed, the date fault tolerance of the SCGC protocol is better than the Growth codes protocol. In total 1000 rounds, the number of symbols recovered is greater, and the time taken to recover 80% symbols is less in the SCGC protocol.

Noted worthy, in the SCGC protocol, the received codewords in sink are forwarded by cache nodes. When a cache node is failed, the sink will employ other neighbor sensor node for replacement. If the disaster occurs multiple times around the sink, most of the cache nodes may failed, leading a sharp decline of data collection efficiency. However, this situation will also cause the failure of data recovery in the Growth codes protocol.

3) NODE ENERGY CONSUMPTION COMPARISON

The data collection efficiency should be improved without affecting the nodes in network. If the energy consumption of the cache nodes is much higher, they would be failed early, which will destroy the stability of the network and shorten the lifetime.

We count the energy consumption various different type of nodes once 50 symbols have recovered in sink during the whole data collection process. Fig. 16 depicts the energy consumption ratio of the cache node and the normal sensor node.

During the whole data collection process, the energy consumption of the cache nodes is lower than the normal sensor nodes. This is because the sensor nodes need to exchange codewords with neighbor in each round while the cache nodes only forward codewords in scheduled round in TDSP. Although the cache nodes need to transmit some special packets which have bigger size than codeword packet, the number of these packets is much smaller than the latter. Therefore, the total energy consumption of cache nodes is smaller than sensor nodes, ensuring the network stability.

In addition, unlike sensor nodes, cache nodes need space to store filter-bitmap, codeword distance list, and forwarding order list. But with the length constraints of update cycle, these space consumptions can be well limited. It should also be noted that, although the cache nodes need to calculate the codeword distance and update information, since the consumption of executing computer instructions is far less than transmission, the neglect is acceptable.

C. APPLICATIONS

Swarm robotics can complete the tasks that are difficult to do with single robot through simple rules and local interactions. It may be a more economical and safer choice to adopt mobile robots to execute surveillance tasks than employing human guards [4]. Since the coordination based on high-quality of information exchange among sensing robots is important in wireless mobile robot swarms, a well-designed data collection strategy with high efficiency is critical. Especially the monitor environments are complex and unstable in the wireless multiple access network. In order to achieve the objectives of system control, it is required to provide correct data transmission within a limited time to ensure data reliability. The SCGC protocol can guarantee the service quality with the high data recovery efficiency. With the SCGC
protocol, interaction between individual sensing robots can be guaranteed well to maintain the cooperative task solving capability in wireless mobile swarm robotics.

VII. CONCLUSION

Swarm robotics is an approach to collective robotics with which it is easily possible to complete the tasks that are difficult to do with single robot. As mobile robots are less likely to be connected via wires, a well coordination between mobile robots which based on data collection is vital in wireless swarm robotics systems, especially in disaster scenarios. Since in the wireless multiple access network, the unpredictable impacts may fail the sensing robots or nodes, causing the loss of valuable data, we proposed a sectional configured data collection strategy based on the Growth codes, the SCGC protocol, from the perspective of reducing the duplicate copies of the recovered symbols. Compared with the zero-configuration network, the neighbor nodes of the sink are set as cache nodes. In this network model, the stable and fast information exchange between the sink and the cache node can be used to filter nearby codewords. Using the information of recovered symbols from the sink, the cache nodes filter and throw the redundant codewords from the forwarding queue to sink. Further, the sink can schedule the forwarding order of the cache nodes, which improves the data recovery efficiency. We also balance the information update overhead with the codeword filtering effect in the cache nodes. The simulation results show that the SCGC protocol can effectively improve data collection efficiency in common random networks and disaster scenarios. In addition, the energy consumption of cache nodes is well controlled to ensure the stability of the wireless multiple access network.

Although the SCGC protocol can improve the overall data collection efficiency, since only the nodes around the sink are scheduled and controlled while other sensor nodes are still exchanging codewords randomly, so the performance in sparse networks is not as expected. In addition, the constraints on the number of cache nodes and the information update cycle still need to be improved, which is a part of our future works.

REFERENCES

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, “Wireless sensor networks: A survey,” Comput. Netw., vol. 38, no. 4, pp. 393–422, 2002.
[2] M. Magno, D. Boyle, D. Brunelli, B. O’Flann, E. Popovic, and L. Benini, “Extended wireless monitoring through intelligent hybrid energy supply,” IEEE Trans. Ind. Electron., vol. 61, no. 4, pp. 1871–1881, Apr. 2014.
[3] B. Zhang, Y. Wu, X. Yi, and X. Yang, “Joint communication-motion planning in wireless-connected robotic networks: Overview and design guidelines,” 2015, arXiv:1511.02299. [Online]. Available: https://arxiv.org/abs/1511.02299
[4] Y. Wu, B. Zhang, S. Yang, X. Yi, and X. Yang, “Energy-efficient joint communication-motion planning for relay-assisted wireless robot surveillance,” in Proc. IEEE Conf. Comput. Commun. (INFOCOM), May 2017, pp. 1–9.
[5] D. A. Patil, M. Y. Upadhye, F. S. Kazi, and N. M. Singh, “Multi robot communication and target tracking system with controller design and implementation of SWARM robot using Arduino,” in Proc. Int. Conf. Ind. Instrum. Control (ICIC), May 2015, pp. 412–416.
[6] W. Han, Z. Tian, Z. Huang, D. Huang, and Y. Jia, “Quantitative assessment of wireless connected intelligent robot swarms network security situation,” IEEE Access, vol. 7, pp. 134293–134300, 2019.
[7] M. Luhy, “LT codes,” in Proc. 43rd Annu. IEEE Symp. Found. Comput. Syst., Jun. 2003, pp. 271–280.
[8] A. Shokrollahi, “Raptor codes,” IEEE Trans. Inf. Theory, vol. 52, no. 6, pp. 2551–2567, Jun. 2006.
[9] G. B. Shankar, G. Singh, S. Bose, and W.-D. Zhong, “Multicasting in wireless networks using rateless codes and opportunistic routing,” in Proc. 10th Int. Conf. Inf., Commun. Signal Process. (ICICS), Dec. 2015, pp. 1–5.
[10] L. Sun, P. Ren, Q. Du, and Y. Wang, “Fountain-coding aided strategy for secure cooperative transmission in industrial wireless sensor networks,” IEEE Trans. Ind. Inform., vol. 12, no. 1, pp. 291–300, Feb. 2016.
[11] Y. Lin, B. Liang, and B. Li, “Priority random linear codes in distributed storage systems,” IEEE Trans. Parallel Distrib. Syst., vol. 20, no. 11, pp. 1653–1667, Nov. 2009.
[12] Y. Suh, J. Baik, N. Rahnavard, and J. Heo, “Fountain code design for broadcasting systems with intermediate–state users,” IEEE Trans. Commun., vol. 63, no. 9, pp. 3057–3068, Sep. 2015.
[13] K. Lei, S. Zhong, F. Zhu, K. Xu, and H. Zhang, “An NDN IoT content distribution model with network coding enhanced forwarding strategy for 5G,” IEEE Trans. Ind. Informat., vol. 14, no. 6, pp. 2725–2735, Jun. 2018.
[14] H. Wang, L. Shao, M. Li, B. Wang, and P. Wang. “Estimation of clock skew for time synchronization based on two–way message exchange mechanism in industrial wireless sensor networks,” IEEE Trans. Ind. Informat., vol. 14, no. 11, pp. 4755–4765, Nov. 2018.
[15] A. Beimel, S. Dolev, and N. Singer, “RT oblivious erasure correcting,” IEEE/ACM Trans. Netw., vol. 15, no. 6, pp. 1321–1332, Dec. 2007.
[16] A. Hagedorn, S. Agarwal, D. Starobinski, and A. Trachtenberg, “Rateless coding with feedback,” in Proc. IEEE 28th Conf. Comput. Commun. (INFOCOM), Apr. 2009, pp. 1791–1799.
[17] M. Hashemi, Y. Cassuto, and A. Trachtenberg, “Fountain codes with nonuniform selection distributions through feedback,” IEEE Trans. Inf. Theory, vol. 62, no. 7, pp. 4054–4070, Jul. 2016.
[18] A. Kamra, V. Misra, J. Feldman, and D. Rubenstein, “Growth codes: Maximizing sensor network data persistence,” ACM SIGCOMM Comput. Commun. Rev., vol. 36, no. 4, pp. 255–266, 2006.
[19] N. Thomos, R. Pulikkoonattu, and P. Frossard, “Growth codes: Intermediate performance analysis and application to video,” IEEE Trans. Commun., vol. 61, no. 11, pp. 4710–4721, Nov. 2013.
[20] 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.
[21] M. Albano and J. Gao, “In–network coding for resilient sensor data storage and efficient data mule collection,” in Proc. Int. Symp. Algorithms Exp. Sensor Syst. Wireles Netw. Distrib. Robot., 2010, p. 105–117.
[22] 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.
[23] D. Yang, Y. Xu, H. Wang, T. Zheng, H. Zhang, H. Zhang, and M. Gilidun, “Assignment of segmented slots enabling reliable real–time transmission in industrial wireless sensor networks,” IEEE Trans. Ind. Electron., vol. 62, no. 6, pp. 3966–3977, Feb. 2015.
[24] A. Kamra, J. Feldman, V. Misra, and D. Rubenstein, “Data persistence for zero-configuration sensor networks,” in Proc. ACM Special Interest Group Data Commun. (SIGCOMM), 2006.
[25] J. Zhang, W. Li, D. Cui, X. Zhao, and Z. Yin, “The NS2-based simulation and research on wireless sensor network route protocol,” in Proc. 5th Int. Conf. Wireless Commun., Netw. Mobile Comput., Sep. 2009, pp. 1–4.

WEI ZHANG received the B.E. degree from the School of Information Science and Engineering, Wuhan University of Science and Technology, China, in 2000, and the M.Ec. and Ph.D. degrees from the Computer School, Wuhan University, China, in 2004 and 2008, respectively. He is currently an Associate Professor with the School of Computer Science and Technology, Hangzhou Dianzi University, China. His research interests include wireless sensor networks and intelligent computing. He is a member of the Association for Computing Machinery and the China Computer Federation.
YUWEI HU received the B.S. degree from the School of Computer Science and Technology, China University of Mining and Technology, in 2017. She is currently pursuing the M.S. degree with the School of Computer Science and Technology, Hangzhou Dianzi University.

YONGJIAN REN received the Ph.D. degree in computing application technology from Florida Atlantic University, in 1998. He is currently a Professor in software engineering with Hangzhou Dianzi University, China. His research interests include cloud computing and mass storage.

YANGYANG BAI received the bachelor’s degree from Shandong Jianzhu University, in June 2018. He is currently pursuing the degree in data transmission reliability with Hangzhou Dianzi University.