RESEARCH ARTICLE

Optimal-cost repair in multi-hop distributed storage systems with network coding
Majid Gerami1*, Ming Xiao1, Mikael Skoglund1, Kenneth W. Shum2 and Dengsheng Lin3

1 Communication Theory Laboratory, School of Electrical Engineering, The Royal Institute of Technology (KTH), Stockholm, Sweden
2 Institute of Network Coding, The Chinese University of Hong Kong, Shatin, NT, Hongkong
3 National Key Lab of Communication, University of Electronics Science and Technology of China, Chengdu, China

ABSTRACT

We study the transmission cost of repair in a distributed storage system, where storage nodes are connected together through an arbitrary network topology, and there is a cost in the use of the network link. Contrary to the classical model, where there exists a link between a pair of storage node, in our repair model there might not exist a link between some pairs of storage nodes or it might be expensive to use. For that, we propose surviving nodes cooperation in repair, meaning that the surviving nodes as the intermediate nodes combine their received packets with their own stored packets and then transmit coded packets towards the new node. We show that surviving node cooperation can reduce the repair-cost, the sum of the costs for transmitting repairing data between the surviving nodes and the new node. For the system that allows surviving node cooperation, we find the minimum-cost codes in repair by firstly deriving a lower bound of the repair-cost through an optimization problem and then proposing achievable codes. We show the gain of the proposed codes in reducing the repair-cost in some scenarios. Copyright © 2016 John Wiley & Sons, Ltd.

*Correspondence
Majid Gerami, Communication Theory Laboratory, School of Electrical Engineering, The Royal Institute of Technology (KTH), Stockholm, Sweden.
E-mail: gerami@kth.se
Received 6 March 2016; Revised 18 July 2016; Accepted 18 July 2016

1. INTRODUCTION

The study on the role of coding in distributed storage systems has recently attracted a lot of research interest. Because of the vast applications of these systems, for example, in data-centres, peer-to-peer networks and wireless sensor networks (WSNs), it will be interesting to investigate the advantages and the costs of coding in these systems. In the recent work, in [1], the advantage of erasure codes in providing higher reliability compared with replication has been presented, while both schemes use the same storage space. In [2–11] erasure codes, based on Reed–Solomon codes, parity array codes and LDPC codes, have been designed for some applications, for example, for HDFS RAID systems in Facebook [5]. In [12, 13] random network coding is proposed for generating redundancy in Wuala peer-to-peer networks. Among the codes, maximum distance separable (MDS) codes1 offer the highest reliability in the use redundancy.

However, the aforementioned advantage of coding does not come free and coding compared with replication may impose higher costs to the storage systems in some scenarios, for example, in repair. In distributed storage systems, when a node fails to maintain the reliability of the system, a new node is generated. The process of generating the new node is termed as the repair process. Recently, the costs in repair has been studied from different aspects, for example, in repair bandwidth [14], the number of disk I/O reads [15], and repair locality [16]. Dimakis et al. in [14] studied the required number of bits in repair, denoted as repair bandwidth and derived the minimum repair bandwidth. A new class of erasure codes, namely, regenerating codes based on network coding ([17, 18]), are proposed in [14, 19]. In the proposed codes, the new node may not have the same encoded data as the failed node; however, the new node and the surviving nodes still preserve the property that every k nodes can reconstruct the original file. This kind of repair is termed as functional repair. The exact regenerating of a new node has been studied in [20, 21]. Because the proposed approach in [14] uses network coding to achieve the optimal repair-bandwidth codes, it requires multiple reads from any storage node. To reduce the number of

1 File size $M$ bits when coded by an $(n,k)$-MDS code, it is divided to $k$ parts and then encoded to $n$ parts, such that any part has $M/k$ bits and any subset of $k$ parts can reconstruct the original file.
reads, fractional repetition codes has been introduced by El Rouayheb and Ramchandran in [15]. Another important criterion is the number of surviving nodes that are connected in repair, denoted as the repair locality. The repair locality has been studied by Papailiopoulos and Dimakis in [16]. Finally, another important criterion in repair is the transmission cost. We study the transmission cost in the repair in a distributed storage system whose nodes are connected through an arbitrary network topology and network links have different costs.

References [14–16, 19, 20, 22] have well-addressed the optimal repair from different aspects. However, the link cost (transmission cost in channels) and the impact of the network topology have not been considered. In a practical system, the transmission cost is an important design consideration and different links (channels) of a network may have different costs. For instance, in data-centres, data transmission between different pairs of storage nodes may have different costs because of the different paths data passes through switches/routers [23, 24]. The topology of a network may also impact the repair cost. Recently, Reference [25] considers the transmission cost from the surviving nodes to the new node, and the cost-bandwidth trade-off is derived. Yet, this model does not exploit the network topology in repair. In recent works, Martalo et al. in [12, 13] studied random network coding in overlay architecture of Wuala networks and showed the benefits of random network coding in generating the new fragments. Yet, the authors in [12, 13] did not formulate an optimization problem over a general network topology to minimize the repair-cost. We address the impact of network topology and transmission cost in repair. In summary, the main contributions of this paper are

- We study the repair process while we consider the network topology and the transmission costs between nodes. In such a system, we define the sum of the costs of transmitting packets between all pairs of nodes in repair as the repair-cost and then we investigate the minimum-cost repair.
- We propose an algorithm in which the optimal repair-cost for an arbitrary network is derived.
- We propose surviving node cooperation (SNC) method and show that it can reduce the repair-cost.
- An upper bound of the finite field size for constructing the optimal codes is derived.
- We study the impact of network topology in repair.

The remainder of the paper is organized as follows. In Section 2, we formulate an optimization problem that offers a lower bound of the repair-cost. Further, in Section 3, we show that the lower bound is achievable, and we characterize the sufficient finite field size for the codes that achieve the bound. In Section 4, we apply the proposed algorithm in large scale distributed storage systems. Finally, in Section 5, we conclude the paper.

2. PROBLEM FORMULATION

Before we formally state the problem, we give a motivating example. Consider a distributed storage system in Figure 1. The system stores a file containing two fragments \( a, b \). The stored fragments are encoded by a \((3, 2)\)-MDS code. That is, every two storage nodes can reconstruct the original file. In this example, node 3 fails, and the new node is generated by the help of nodes 1 and 2. Assume, transmitting a fragment from nodes 1 and 2 to the new node costs 10 units and transmitting a fragment from node 1 to node 2 costs 1 units. In practice, these various costs can come from the fact that nodes 1 and 2 are in the same storage rack, while the new node located in another rack. In classical repair model, nodes 1 and 2 send fragments \( a \) and \( b \) to the new node to generate the fragment \( a + b \) in the new node. This costs \( 10 + 10 = 20 \) units. When surviving node can cooperate in repair, node 1 sends the fragment \( a \) to node 2. Node 2 then combines its received fragment with its stored fragment and generate the fragment \( a + b \) and sent it to the new node. The repair with surviving node cooperation costs \( 1 + 10 = 11 \) units. This shows that surviving node cooperation can reduce the repair-cost from 20 units to 11 units, in this example.

Earlier, for a specific network, we have studied the repair-cost and proposed coding at the intermediate nodes to reduce the cost. A natural question is that what the optimal repair-cost is and how to design codes that achieve the optimal point for more general scenarios. In this section, we formulate an optimization problem that establishes a fundamental lower bound on the repair-cost for an arbitrary network.

2.1. System model

Consider a storage system with the original file of size \( M \) (measured in fragments) distributed among \( n \) nodes in which each node stores \( \alpha \) fragments and any \( k \) of \( n \) nodes can rebuild the original file. We focus on the distributed storage system that stores the file by an MDS code. We denote the source file by an \( M \times 1 \) vector, \( s \). Vector \( s \) consists of elements from a finite field \( F_q \), where \( q \) is the finite field size. Then, the code on each node \( i \) can be represented by

Figure 1. Surviving node cooperation can reduce the repair cost.
a matrix \(Q_t = (q_1^t, \ldots, q_n^t) \in \mathbb{F}_M^{M \times n}\), where each column \((q_j^t)\) represents the code coefficients of fragment \(j\) on node \(i\). Thus, the coded data in node \(i\) is \(x_i = Q_t^T s\).

Next, we model the information flow in a repair process by a directed acyclic graph \(G(N, A)\), where \(N\) is the set of nodes and \(A\) is the set of directed links. The information flow graph consists of three different nodes: a source node, storage nodes and several data collectors (DCs). The source node distributes the original file among storage nodes by the (presumably) infinite-capacity links. Every storage node can be denoted by input (in) and output (out) nodes connecting by a link of capacity \(a\). Finally, DC reconstructs the original file by connecting to at least \(k\) storage nodes via infinite-capacity links and then recover the stored file from \(k\) linear equations by, for example, Gaussian elimination method. In contrast to the classical repair model [14], there might not exist links between some of the surviving nodes and the new node, and information from surviving nodes passes a number of intermediate nodes to reach to the new node. Meanwhile, the intermediate nodes are allowed to transmit codewords, which are functions of their received fragments and their stored fragments. We assume intermediate nodes are capable of performing linear operations in the finite fields. The number of surviving nodes in repair, which is denoted as \(d\), is assumed to be greater than \(k\), that is, \(d \geq k\) (note that repair for \(d < k\) is information-theoretically impossible). In our model, there is a cost associated with edge \((ij)\), for \((ij) \in A\). The cost of link \((ij)\), which is denoted as \(c_{ij}\), represents the link cost for transmitting one fragment from node \(i\) to node \(j\). We note that \(c_{ij}\) is a finite and non-negative real number. We assume all the fragments have equal size and then only consider linear costs.\(^3\) This means if the transmission cost of one fragment from node \(i\) to \(j\) is \(c_{ij}\), then it costs \(mc_{ij}\) to transmit \(m\) fragments from node \(i\) to \(j\). An information flow graph for (one stage of) repair on node 4 in the four-node tandem storage network has been shown in Figure 2.

### 2.2. Repair-cost formulation

To study the repair-cost, we define vector \(z = [z_{(ij)} : (ij) \in A]^T\), where each element \(z_{(ij)}\) is a non-negative integer number denoting the number of fragments sent from node \(i\) to \(j\) in the repair. Accordingly, we define vector \(c = [c_{ij} : (ij) \in A]^T\). Hence, the repair-cost, denoted by \(\Gamma\), is computed by

\[
\Gamma \triangleq c^T z
\]  

Our objective is to minimize the repair-cost \((\Gamma)\) in which the system with the new node retains the MDS property. To examine the MDS property of the system after each stage of repair, DC contacts the new node and a set of \(k - 1\) surviving nodes among \(n - 1\) storage nodes. This yields \((n - 1)\) constraints on one stage of repair. We note that, potentially, there are infinite stages of repair. That means after generating a new node, again a node may fail. There, the previously regenerated node (if still alive) can help a currently new node to be generated. Consequently, the network is evolving after each stage of repair. Depending on the network topology and the link costs, the network can evolve for infinite cases. This can make the analysis for a general network complicated. We instead derive a lower bound of repair by the cut-set bound analysis only on the first stage of repair. Then, in the next section, we prove that this bound is achievable for MDS codes.

Thus, we minimize the repair-cost (denoted as \(\Gamma\)) under the constraints that all the cuts of connecting DC to the new node and \(k - 1\) other storage nodes must be greater than or equal to \(M\), the original file size. For instance, in Figure 2, the heavy dotted line illustrates a cut when DC connects to the new node and node 1 for an MDS code with \(k = 2\). The cut constraint relating to this cut can be formulated by the inequality: \(z_{35} + \alpha \geq M\). By assuming vector \(z = [z_{(12)}z_{(23)}z_{(35)}]^T\), we can express the inequality in a vector space as, \((0, 0, 1)z \geq M - \alpha\). Denote \(r\) as the total number of cut constraints when DCs connect to the new node and a set of \(k - 1\) out of \(n - 1\) surviving nodes. For each set of \(k - 1\) surviving selection, if the min-cut on the network is greater than \(M\), then DC have access to the file. Then, \(r = \binom{n-1}{k-1}\). Denoting \(|A|\) as the cardinality of existing edges between nodes, we form all the inequalities in a matrix form, by defining an \(r \times |A|\) dimensional matrix \(L\) (this matrix is called coefficient matrix [26]). The corresponding inequalities induced by the cut constraints show a region in a multi-dimensional space that the subgraph must satisfy to be a feasible solution. This region is often called polytope [26]. Consequently, the polytope is

\[
\Psi = \{z = [z_{(ij)}] \mid z_{(ij)} \geq 0, Lz \geq b\}
\]  

where the comparison of two vectors, for example, \(a \geq b\) means every element in \(a\) is greater than or equals to the element in \(b\) at the same position.

**Example 1.** In Figure 2, if DC connects to node 1 and the new node to rebuild the source \((k = 2)\), the first cut constraint is

\[
z_{(35)} \geq M - \alpha
\]  

The second constraint follows if we connect DC with node 2 and the new node. Hence,

\[
z_{(12)} + z_{(35)} \geq M - \alpha
\]  

Finally, when DC connects to node 3 and the new node, we have the third constraint:

\[
z_{(23)} \geq M - \alpha
\]
Thus, we can form all these inequalities in a matrix form as

$$
\begin{bmatrix}
0 & 0 & 1 \\
1 & 0 & 1 \\
0 & 1 & 0 \\
\end{bmatrix}
\begin{bmatrix}
z_{(12)} \\
z_{(23)} \\
z_{(35)} \\
\end{bmatrix}
\geq
\begin{bmatrix}
M - \alpha \\
M - \alpha \\
M - \alpha \\
\end{bmatrix}
$$

(6)

Because the constraint region and the objective function in the repair problem are linear, and $z_{(ij)}$s are integer numbers then the problem is an integer linear programming problem.

$$
\text{minimize } \Gamma = c^Tz \\
\text{subject to } Lz \geq b, \\
z_{(ij)} \in \mathbb{Z}^+ 
$$

(7)

where $\mathbb{Z}^+$ is the set of non-negative integers. An integer linear programming problem has in general high complexity. We apply a relaxation technique by assuming that $z_{(ij)}$s are real numbers. Later, we argue that the loss of optimality by following the aforementioned assumption is small if the file size is large. This relaxation transforms the integer linear programming problem into a linear programming problem, as

$$
\text{minimize } \Gamma = c^Tz \\
\text{subject to } Lz \geq b, \\
z_{(ij)} \in \mathbb{R}^+ 
$$

(8)

where $\mathbb{R}^+$ is the set of non-negative real numbers. This problem can be solved efficiently, because it has a polynomial-time complexity in number of nodes. Because of the applied relaxation, the aforementioned linear programming problem gives a lower bound of the repair-cost.

Example 2. Consider a four-node distributed storage system in the tandem network, as shown in Figure 3. We assume three nodes are cooperating in the repair process $(d = 3), M = 4, k = 2, \alpha = 2, z = [z_{(12)}z_{(23)}z_{(35)}]$ and the corresponding cost vector is

$$
c = (c_{(12)} c_{(23)} c_{(35)})
$$

(9)

In other words, there exists a link between nodes 1 and 2 having cost $c_{(12)}$ units for transmitting one fragment, and costs $c_{(23)}$ and $c_{(35)}$ units for transmitting one fragment respectively from nodes 2 to 3, and from nodes 3 to 5 (new node). Now we analyse the constraint region for $M = 4, \alpha = 2$ in (6). Hence, we can formulate the problem as

$$
\text{minimize } \Gamma(z) = c_{(12)}z_{(12)} + c_{(23)}z_{(23)} + c_{(35)}z_{(35)} \\
\text{subject to } 
\begin{bmatrix}
0 & 0 & 1 \\
1 & 0 & 1 \\
0 & 1 & 0 \\
\end{bmatrix}
\begin{bmatrix}
z_{(12)} \\
z_{(23)} \\
z_{(35)} \\
\end{bmatrix}
\geq
\begin{bmatrix}
2 \\
2 \\
2 \\
\end{bmatrix}
$$

(10)

Solving the linear optimization problem (e.g. by the simplex method [26]) gives the optimal subgraph $(z_{(12)},z_{(23)},z_{(35)}) = (0, 2, 2)$ with the cost of $2c_{23} + 2c_{35}$ units.

A linear network coding by selecting coefficients (in this example) from $\mathbb{F}_5$ with SNC in Figure 3 can meet the minimum-cost subgraph. The coding scheme is by transmitting fragments $p_2 = 2a_1 + b_2, p_3 = a_2 + 2b_2$ from nodes 2 to 3, and then transmitting fragments $p_4 = p_2 + (a_1 + b_1 + a_2 + b_2) = a_1 + b_1 + 3a_2 + 2b_2$ and $p_5 = p_3 + (a_1 + 2b_1 + a_2 + 2b_2) = a_1 + 2b_1 + 2a_2 + 4b_2$ to the new node. Here, $p_4$ and $p_5$ are fragments for the new node. The new node along with the surviving nodes satisfy the MDS property (every $k$ nodes can reconstruct the original file).
In the previous example, the linear optimization gave the exact optimal solution. That was due to the appropriate selection of the file size, $M$, such that the linear program outputs integer values for $z_{ij}$. In general, we can state the accuracy of the our proposed method in the following proposition.

**Proposition 1.** The linear programming problem in (8) yields the (exact) optimal-cost repair for large $M$.

**Proof.** It is known that if the linear programming problem has a finite optimal solution, then the $z_{ij}$s are rational numbers. Because for the optimal-cost problem, there is a finite solution (e.g. transmitting all the file to the new node is a feasible solution with a finite cost) then $z_{ij}$s are rational number. If the $z_{ij}$s of the optimal solution is rational, then we can make the resulted $z_{ij}$s integer by a proper scaling of the file size. For instance, if for $M = 10$ (measured in fragments), the optimization problem gives $z_{ij} = 0.1$, then we scale $M$ to $M = 100$. Then, we would have $z_{ij} = 1$. To be able to scale $M$, the file size, measured in bits, should be large enough. This is almost always the case for the files stored in large scale data-centres.

**Remark 1.** We note that in the optimal repair-cost problem, $d$ number of surviving nodes are willing to help regenerating the new node; however, the optimal repair-cost policy may not use all the surviving nodes in the repair, for the matter of costs. For instance, in the aforementioned example, we have $d = 3$, but the cooperation of only two nodes (nodes 2 and 3) gives the optimal repair-cost.

**Remark 2.** We can relate the repair-cost to the consumed energy or bandwidth in repair. Assume that transmitting $z_{ij}$ fragments from nodes $i$ to $j$ requires $c_{ij}$ units of energy. Then transmitting $c_{ij}z_{ij}$ fragments from nodes $i$ to $j$ requires $c_{ij}z_{ij}$ units of energy. By this formulation, $\sum c_{ij}z_{ij}$ is the total energy consumption in repair. Then, the linear programming problem minimizes the energy consumption in repair. We can also relate the repair-cost to bandwidth. When $z_{ij}$ is the number of bits transmitted from nodes $i$ to $j$, then $\sum z_{ij}$ denotes the total bandwidth in repair. Then, the linear programming problem minimizes the required bandwidth in repair.

### 3. Achievable Codes for Minimum Repair-Cost

In what follows, we shall show that the lower bound of the repair-cost is achievable by a linear code and for $\alpha = M/k$. That is, there exists a linear code corresponding to the repair with the minimum-cost subgraph from the optimization problem (8). Our proof is based on random linear codes and then we discuss the required finite field size for constructing the minimum repair-cost MDS codes. To find the sufficient field size for successful regeneration, we apply sparse-zero lemma as follows.

**Lemma 1.** Consider a multi-variable polynomial $g(\alpha_1, \alpha_2, \ldots, \alpha_n)$ which is not identically zero, and has the maximum degree in each variable at most $d_0$. Then, there exist variables $\gamma_1, \gamma_2, \ldots, \gamma_n$ in the finite field $\mathbb{F}_q$, and $q \geq d_0$, such that $g(\gamma_1, \gamma_2, \ldots, \gamma_n) \neq 0$.

**Proof.** See proof of Lemma 19.17 in [27].

Suppose a source information file, containing $M$ fragments, is coded by an $(n, k)$-MDS code, that is, each node...
stores $\alpha = M/k$, and every $k$ nodes can reconstruct the original file. If $x_i$ denotes the stored symbols of node $i$, then $x_i = Q^T_s s$, where $Q$ is an $M \times \alpha$-dimensional matrix, which represents the coding coefficients of node $i$. When a node fails (without loss of generality, we assume node 1 fails) solving the optimization problem provides the minimum-cost subgraph. Following the minimum-cost subgraph, the new node is regenerated by the surviving node cooperation. Clearly, with the minimum-cost subgraph, we also know which nodes should encode on the directed information flow graph. Then, by using network codes from a proper finite field, the new node is regenerated. Assume that the coding coefficient matrix of the new node is $Q_1$ and the content of the new node is $x_1$, then $x_1 = Q_1^T s$.

To maintain the MDS property after the repair, the coding coefficient matrix $(Q_1)$ have to meet certain requirement. That is, for any selection of $k-1$ out of $n-1$ surviving nodes, $\Xi_{k-1} = \{|Q_{s_1}, \ldots, Q_{s_{n-1}}|\}$ along with the codes of the new node $Q_1$, the polynomial $\text{det} \left( \left[ Q_1, Q_{s_1}, \ldots, Q_{s_{n-1}} \right] \right)$ is a non-zero polynomial. In what follows, we first show that $\text{det} \left( \left[ Q_1, Q_{s_1}, \ldots, Q_{s_{n-1}} \right] \right)$ satisfying the subgraph of the optimization process is not identically zero and then discuss the required field size.

**Lemma 2.** For regenerating node 1, there exist linear codes satisfying the minimum-cost subgraph (resulted from problem (8)) such that the polynomial $\text{det} \left( \left[ Q_1, Q_{s_1}, \ldots, Q_{s_{n-1}} \right] \right)$ is non-zero for any selected set $\Xi_{k-1}$. That is,

$$\prod_{\{s_1, \ldots, s_{n-1}\} \subseteq \{2, \ldots, n\}} \text{det} \left( \left[ Q_1, Q_{s_1}, \ldots, Q_{s_{n-1}} \right] \right) \neq 0 \quad (11)$$

**Proof.** Consider $\Xi_k = \{|Q_{s_1}, \ldots, Q_{s_k}|\}$ a set of coding coefficients selected from $k$ out of $n$ nodes. Because every $k$ nodes can reconstruct the original file, then the matrix $[Q_{s_1}, \ldots, Q_{s_k}]$ has full rank $M = k\alpha$. Thus, for $\Xi_{k-1}$, the matrix $[Q_1, Q_{s_1}, \ldots, Q_{s_{n-1}}]$ has rank $(k-1)\alpha$. Consider a set $\mathcal{V}$ containing the data collector, $in$ and $out$ nodes of the new node, and $out$ nodes of the nodes in set $\Xi_{k-1}$. Other nodes including the source node are in the complement set $\overline{\mathcal{V}}$. Because all the cuts has capacity $M$ fragments, there would be $R = M - (k-1)\alpha = \alpha$ fragments, the capacity of edges from $out$ nodes of the set $N - \Xi_{k-1}$ to the in node of the new node. In the set $\mathcal{N} = \Xi_{k-1}$, there exist $\alpha$ fragments that have independent vectors from the vectors in $\Xi_{k-1}$. If we send those $\alpha$ fragments (through the links with the total capacity $R = \alpha$) to the new node, then the matrix of the coding coefficients will be full rank. Therefore, $\text{det} \left( \left[ Q_1, Q_{s_1}, \ldots, Q_{s_{n-1}} \right] \right)$ can be non-zero.

To find the required field size of the codes, we need to know the maximum degree of the variables of the polynomial in (11). For analysis, we use $n_{nc}$ to denote the maximum number of encoding processes operated over a fragment in the repair. We note that $n_{nc} \leq |N|$, the number of nodes in the networks.

**Example 1:** In the repair process of node 4 in tandem network (Figure 2), $n_{nc} = 4$. The encoding can be at nodes 1, 2, 3 and at the new node.

**Theorem 1.** For a distributed storage system DSS$(n, k, \alpha)$ with the source file of size $M$, if the field size is greater than $d_0$, there exists a linear code such that the MDS property is satisfied for any stage of repair, where

$$d_0 = \left( \frac{n}{k} \right) M_{n_{nc}} \quad (12)$$

**Proof.** The proof is by induction on the number of repair stages. That is, we assume before a node fails all the storage nodes have the MDS property. In each stage of repair, when a node fails, the new node is regenerated preserving the MDS property. Thus, we initialize the code on $n$ nodes by which any $k$ out of $n$ nodes can reconstruct the original file. Then if a node fails, the new node is regenerated such that the repairing cost is minimized and the MDS property is preserved. By the MDS property, the coding coefficients of any $k$ nodes must have full rank $M$. That is,

$$\prod_{\{s_1, \ldots, s_{n-1}\} \subseteq \{1, \ldots, n\}} \text{det} \left( \left[ Q_{s_1}, \ldots, Q_{s_{n-1}} \right] \right) \neq 0 \quad (13)$$

The maximum degree of variables in (13) is $\left( \begin{smallmatrix} n \\ k \end{smallmatrix} \right) M$. Thus, by Lemma 1, if the field size $q$ is greater than $\left( \begin{smallmatrix} n \\ k \end{smallmatrix} \right) M$, then there is a network coding solution for repair. Because $n_{nc} \geq 2$ (at least two coding process: one in surviving nodes and another in the new node), $d_0 \geq \left( \begin{smallmatrix} n \\ k \end{smallmatrix} \right) M$; thus, there is a coding solution for $q \geq d_0$.

When a node fails (assume $Q_1$), the optimization algorithm finds the minimum-cost subgraph. Accordingly, the fragments are combined using linear network coding, and then the new node is regenerated. The set including the new node ($Q_1$) and surviving nodes must satisfy the MDS property. Thus,

$$\prod_{\{s_1, \ldots, s_{n-1}\} \subseteq \{2, \ldots, n\}} \text{det} \left( \left[ Q_1, Q_{s_1}, \ldots, Q_{s_{n-1}} \right] \right) \neq 0 \quad (14)$$

By Lemma 2, the polynomial can be nonzero. The maximum degree of each variable is less than $\left( \begin{smallmatrix} n-1 \\ k-1 \end{smallmatrix} \right) M_{n_{nc}}$. By Lemma 1, if the finite field size $q \geq \left( \begin{smallmatrix} n-1 \\ k-1 \end{smallmatrix} \right) M_{n_{nc}}$, there is a network solution for the repair. Clearly, $d_0 = \left( \begin{smallmatrix} n \\ k \end{smallmatrix} \right) M_{n_{nc}} \geq \left( \begin{smallmatrix} n-1 \\ k-1 \end{smallmatrix} \right) M_{n_{nc}}$ for $n \geq k$. Hence, for $q \geq d_0$, there exist a code for the repair. This concludes our proof. □

**Remark 3.** The field size in Theorem 1 is an upper bound for the finite field size. In practice, however, the required field size can be considerably smaller. For example, we in our recent work in [28], designed an explicit code for tandem and grid networks that requires than the aforementioned upper bound.

In summary, the minimum repair-cost MDS code is derived by two steps: Firstly, the optimal-cost subgraph is
derived by solving a linear programming problem (8). Secondly, the coded fragments of the new node is regenerated by either random linear coding [29] or deterministic [30] from the finite field size determined by Theorem 1.

4. NUMERICAL RESULTS

In this section, we examine our proposed method on some storage networks and show the gain of our proposed approach in reducing the repair-cost. Studies in this section will also show that the network topology can affect the repair-cost.

4.1. Large scale tandem storage network

A large scale tandem storage network is illustrated in Figure 4. That is, each node is linked to two neighbouring nodes. When a node fails and a new node joins, the repair traffic is encoded and then forwarded by the intermediate storage nodes towards the new node. In a tandem network, the optimal repair-cost is derived by the following proposition.

**Proposition 2.** Consider a tandem distributed storage network where each node stores $M/k$ fragments such that every $k$ nodes can reconstruct the original file of size $M$ fragments. Suppose that $d = k$ surviving nodes help in repair. Assume that the cost of transmitting a fragment from node $i$ to node $j$ is denoted by $c_{ij}$, where $c_{ij}$ is non-negative real number. Then for all values of $c_{ij}$'s, the optimal repair-cost is achieved by cooperation of the $k$ closest surviving nodes to the new node and that is obtained by transmitting $M$ fragments between surviving nodes.

**Proof.** For achievability, assume the source file of a size $M$ is split into $k$ fragments and represented by a vector $s$. We construct an $(n,k)$-MDS code using the following Vandermonde matrix, as a generator matrix, $G$,

$$G = \begin{pmatrix} 1 & \alpha_1 & \alpha_1^2 & \cdots & \alpha_1^{k-1} \\ 1 & \alpha_2 & \alpha_2^2 & \cdots & \alpha_2^{k-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \alpha_n & \alpha_n^2 & \cdots & \alpha_n^{k-1} \end{pmatrix} \quad (15)$$

That is, if we denote the coded fragment on node $i$ by a vector $\mathbf{x}_i$ and the $i$th row on the matrix $G$ by a vector $\mathbf{g}_i$, then $\mathbf{x}_i = \mathbf{g}_i s$. For regenerating the content of node $i$, assume a set of $k_1$ nodes on the left and $k_2$ nodes on the right-hand side of the new node are closest neighbouring nodes, for $k_1 + k_2 = k$. Each node in the set combines its received fragment with its own stored fragment and transmits towards the new node. Finally, a fragment $\eta_1 x_{i-k_1} + \cdots + \eta_k x_{i+k_2}$ is obtained at the new node. Because every $k$ rows of matrix $G$ is full rank, we can always find proper coefficients such that $\eta_1 x_{i-k_1} + \cdots + \eta_k x_{i+k_2} = x_i$. That recovers the content of the new node.

For the converse, we use cut-set bound analysis. Every cut of selecting $k - 1$ storage node and the new node must be greater than $M$. This shows that for every node in the set of $k$ node helping to regenerate the new, they have to transmit greater than $M/k$ fragment on its edge towards the new node.

**Proposition 3.** Consider the previous tandem distributed storage network. If it is allowed to use $d \geq k$ surviving nodes in repair, then for all non-negative values of $c_{ij}$, the optimal repair-cost is still achieved by cooperation of the $k$ closest surviving nodes to the new node and that is obtained by transmitting $M$ fragments between surviving nodes.

**Proof.** See appendix A. □

This is quite a surprising result that in a tandem storage network, only $k$ closest storage nodes are necessary and sufficient for the optimal repair and then increasing number of helper nodes in repair does not benefit the system. This result is in contrast to the results where repair-bandwidth is studied [14], where the repair-bandwidth is a decreasing function of $d$ [14]. This result shows us the effect of the network topology in repair.

4.2. Large scale grid storage network

In a large scale grid storage network, each storage node has four neighbouring nodes. This is illustrated in Figure 5. For this network, when there are $d = k$ surviving nodes in the repair, the optimal repair-cost is derived similar to the tandem storage network, as stated in the following proposition.

**Proposition 4.** Consider a grid distributed storage network where each node stores $M/k$ fragments such that every $k$ nodes can reconstruct the original file. Assuming the cost of transmitting a fragment from node $i$ to node $j$ is denoted by $c_{ij}$, then for all non-negative values of $c_{ij}$, the optimal repair-cost is achieved by cooperation of the $k$ closest surviving nodes and that is obtained by transmitting $M$ fragments between storage nodes.

![Figure 4](image-url) A large scale tandem storage network. The optimal-cost repair is by cooperation of only $k$ storage nodes, and increasing number of helper nodes, $d > k$, does not reduce the repair-cost.
Figure 5. Large scale grid storage network. In this figure, the symbol ‘*’ represents a transmitted fragment in repair when \( d = 4 \). In addition, the symbol ‘+’ represents a transmitted fragment in repair when \( d = 5 \).

Figure 6. The proposed scheme uses surviving node cooperation and outperforms classical repair. For each point on the graphs, we run the experiment for 1000 times, and then calculate the average of repair-cost.

Proof. The proof is similar to the proof in Proposition 2.

Unlike repair in tandem storage network, here by increasing \( d \), the repair cost can be decreased. We show this by the following example.

Corollary 1. In the repair of node \( n_{11} \) in Figure 5 by the help of nodes \( n_{21}, n_{22}, n_{23}, n_{12}, n_{13} \), the optimal-cost repair is seven units corresponding to the minimum-cost subgraph 
\[
(1^{(21)}(11) \cdot 1^{(22)}(12) \cdot 1^{(23)}(13) \cdot 1^{(12)}(13) \cdot 1^{(12)}(11)) = (1, 1, 1, 2, 2).
\]

Proof. See appendix B. □

This shows that increasing the connectivity of storage nodes will reduce the repair-cost, for example, in the preceding example from eight to seven units.

4.3. Fully-connected storage network

In a fully-connected storage network, there exists always a link between a pair of storage nodes. We show by extensive simulations that surviving node cooperation can also reduce the repair-cost in a fully connected network. We first set the parameters in the distributed storage system to \( n = 10, k = 5, M = 5 \). We assume the cost of a link in the network is a random variable that has a uniform distribution over the range \([0, \sigma]\), where \( \sigma \) changes from 1 to 10 units of cost. In this setting, we compare the optimal-cost repair when SNC is allowed (i.e. proposed scheme) with the optimal-cost in classical repair model. To find the optimal point, we use Prim’s algorithm, which has polynomial-time complexity [31]. In Figure 6, we compare the optimal repair-cost in these two schemes over varying \( \sigma \). For each point on the graphs, we perform the experiment for 1000 times and then evaluate the average point. Figure 6 shows that the proposed scheme has always a lower cost compared with the classical repair scheme. This gain of our proposed scheme increases when the link costs vary in a larger range (i.e. increasing \( \sigma \)).

Next, we fix the parameters in the distributed storage system to \( k = 5, M = 5, \sigma = 10 \). We assume that the cost of a link in the network is a random variable that has a uniform distribution over the range \([0, 10]\). In this setting, we compare the optimal-cost repair when SNC is allowed (i.e. proposed scheme) with the optimal-cost in classical repair model. In Figure 7, we compare these two schemes.
For each point on the graphs, we run the experiment for 1000 increases. Therefore, the repair-cost decreases by increasing the number of nodes, the chance of having smaller cost links.

5. CONCLUSIONS

We studied the repair-cost in distributed storage systems where storage nodes are connected together by an arbitrary network topology. We proposed the optimal codes that exploit the multi-hop network structure in repair. We formulated a linear programming problem that gives the fundamental lower-bound of the repair-cost. The linear programming also leads us to the optimal-cost policy for MDS codes. We proved the existence of the code by the random linear code over a large finite field size. We discussed the required field size for the existence of the code. To reduce the cost in networks, we proposed the surviving node cooperation approach. We examined our proposed method over some large-scale storage networks and then presented the advantages in deploying the optimal-cost repair. Studying the repair in networks having the constraint on bandwidth and delay in fragment transmissions is interesting and can be followed as future works.

APPENDICES

Proof of Proposition. 3

Assume \( d \geq k \) surviving nodes are helping a new node to be regenerated. Nodes are labelled from \( d \) to 1 as shown in Figure A1. For the convenience, let us assume that node 1 fails and then consecutive nodes 2 till \( k + 1 \) are the \( k \) closest nodes to the new node (1'), which is located at the same position as node 1. We derive the constraint region in a general form and then find the minimum repair-cost. We obtain the constraint region by the following steps:

Step 1: We connect DC to the new node and nodes \( k - 1 \) other nodes in the set of helper nodes \( \mathcal{H} = \{2, 3, \ldots, d + 1\} \). Then the cut constraint is,

\[
(z_{21'}) + (k - 1)\alpha \geq M \Rightarrow z_{21'} \geq M/k \quad (A1)
\]

Step 2: We connect DC to the new node, node 2 and \( d - 2 \) other nodes in \( \mathcal{H} = \{2, 3, \ldots, d + 1\} \) for \( i \in \{3, \ldots, k + 1\} \). Then the cut that passes the edge \((i, i - 1)\) gives the following constraint:

\[
z_{(i,j-1)} + (k - 1)\alpha \geq M \Rightarrow z_{(i,j-1)} \geq M/k, \text{ for } i \in \{3, \ldots, k + 1\}. \quad (A2)
\]

Step 3: We connect DC to the new node, node \( 2, \ldots, k + 1 \). Then the cut constraint is,

\[
z_{(j,j-1)} + k\alpha \geq M \Rightarrow z_{(j,j-1)} \geq 0
\]

for \( j \in \{k + 2, k + 3, \ldots, d\} \) \quad (A3)

This shows \( k \) closest neighbours must individually transmit \( M/k \) fragments towards the new node. Hence, whatever the value of \( d \) is, the \( k \) closest neighbours to the new node must totally transmit \( k \times M/k = M \) fragments, which poses the cost of \( \sum_{i=1}^{k+1} z_{(i,j-1)} M/k \) units in repair. Thus, increasing \( d \) more than \( k \) does not reduce the repair-cost.

Proof of Corollary 1

The corresponding cost vector \( \mathbf{c} \) for the repair on node 6 is as below. We formulate the linear optimization problem as follows. There are \( \binom{6}{4} = 10 \) cut constraints. Figure A2 shows one of these cut constraints. For convenience, let call nodes \( n_{21}, n_{22}, n_{23}, n_{13}, n_{12}, n_{11} \) respectively as nodes 1,2,3,4,5,6. Then we have,

\[
\mathbf{c} = [c_{(12)}c_{(23)}c_{(14)}c_{(25)}c_{(36)}c_{(56)}]^T = \mathbb{1} \quad (A4)
\]

\[
\min \quad \mathbf{z}^T \mathbf{y} \quad \text{s.t. } \mathbf{Hz} \geq (M - 3\alpha) \mathbb{1} \\
\quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \qu
Figure A1. A cut analysis in a tandem network. The cut analysis in this figure corresponds to the constraint $z_{21} + (k - 1)\alpha \geq M$.

Figure A2. A cut analysis in the $2 \times 3$ grid network with $n = 6$, $k = 4$. The cut in this figure corresponds to the constraint $z_{(66)} \geq M - 3\alpha$.

mentioned in Figure A2 constructs the 8th row in matrix $H$, corresponding to inequality $z_{(66)} \geq M - 3\alpha$.

Solving this linear optimization problem (e.g. by the simplex method) for the $M = 8$ and $\alpha = 2$ results: $\Gamma = 7$, and $z = (z_{(12)}, z_{(14)}, z_{(23)}, z_{(25)}, z_{(36)}, z_{(45)}, z_{(56)}) = (0, 1, 0, 1, 1, 2, 2)$

REFERENCES

1. Weatherspoon H, Kubiatowicz JD. Erasure coding vs. replication: A quantitative comparison. In Peer-to-Peer Systems. Springer: Berlin, Germany, 2002; 328–337.

2. Plank J.S. Erasure codes for storage applications. In Proceeding of the 4th USENIX Conference on File and Storage Technologies (FAST), San Francisco, USA, 2005; 1–74.

3. Rhea SC, Eaton PR, Geels D, Weatherspoon H, Zhao BY, Kubiatowicz J. Pond: The oceanstore prototype. In FAST, 2003; 1–14.

4. Dabek F, Li J, Sit E, Robertson J, Kaashoek MF, Morris R. Designing a DHT for low latency and high throughput. In NSDI, Vol. 4, 2004; 85–98.
5. Sathiamoorthy M, Asteris M, Papailiopoulos D, Dimakis A.G, Vadali R, Chen S, Borthakur D. Xoring elephants: Novel erasure codes for big data. In *Proceeding of the 39th International Conference on Very Large Data Bases, VLDB Endowment*, Riva del Garda, Italy, 2013; 325–336.

6. Bhagwan R, Tati K, Cheng Y, Savage S, Voelker GM. Total recall: System support for automated availability management. In *NSDI*, Vol. 4, 2004; 25–25.

7. Sathiamoorthy M, Dimakis AG, Krishnamachari B, Bai F. Distributed storage codes reduce latency in vehicular networks. *IEEE Transaction on Mobile Computing* 2014; 13(9): 2016–2027.

8. Hafner JL. Weaver codes: Highly fault tolerant erasure codes for storage systems. In *Fast*, Vol. 5, 2005; 16–16.

9. Huang C, Xu L, Star: An efficient coding scheme for correcting triple storage node failures. *IEEE Transactions Computers* 2008; 57(7): 889–901.

10. Xu L, Bruck J. X-code: Mds array codes with optimal encoding. *IEEE Transactions on Information Theory* 1999; 45(1): 272–276.

11. Blaum M, Brady J, Bruck J, Menon J. Evenodd: An efficient scheme for tolerating double disk failures in raid architectures. *IEEE Transactions on Computers* 1995; 44(2): 192–202.

12. Martalo M, Picone M, Bussandri R, Amoretti M. A practical network coding approach for peer-to-peer distributed storage. In *Proceedings of IEEE International Symposium on Network Coding (NETCOD)*, 2010; 1–6.

13. Martalo M, Picone M, Amoretti M, Ferrari G, Raheli R. Proceedings of Information Theory and Applications Workshop, San Diego, USA, 2011; 1–7.

14. Dimakis AG, Godfrey PB, Wu Y, Wainwright MJ, Ramchandran K. Network coding for distributed storage systems. *IEEE Transactions on Information Theory* 2010; 56(9): 4539–4551.

15. El Rouayheb S, Ramchandran K. Fractional repetition codes for repair in distributed storage systems. In *Proceedings 48th Annual Allerton Conference*, Allerton, USA, 2010; 1510–1517.

16. Papailiopoulos DS, Dimakis AG. Locally repairable codes. *IEEE Transactions on Information Theory* 2014; 60(10): 5843–5855.

17. Ahlswede R, Cai N, Li SY, Yeung RW. Network information flow. *IEEE Transactions on Information Theory* 2000; 46(4): 1204–1216.

18. Koetter R, Méard M. An algebraic approach to network coding. *IEEE/ACM Transaction on Networking* 2003; 11(5): 782–795.

19. Wu Y. Existence and construction of capacity-achieving network codes for distributed storage. *IEEE Journal on Selected Areas in Communications* 2010; 28(2): 277–288.

20. Rashmi KV, Shah NB, Kumar PV, Ramchandran K. Explicit construction of optimal exact regenerating codes for distributed storage. In *Proceeding of Allerton Conference Communication, Control, and Computing*, Allerton, USA, 2009; 1243–1249.

21. Shah NB, Rashmi KV, Kumar PV, Ramchandran K. Distributed storage codes with repair-by-transfer and nonachievability of interior points on the storage-bandwidth tradeoff. *IEEE Transactions on Information Theory* 2012; 58(3): 1837–1852.

22. Hu Y, Xu Y, Wang X, Zhan C, Li P. Cooperative recovery of distributed storage systems from multiple losses with network coding. *IEEE Journal on Selected Areas 2010; 28(2): 268–276.

23. Raiciu C, Barre S, Pluntke C, Greenhalgh A, Wischik D, Handley M. Improving datacenter performance and robustness with multipath tcp. In *ACM Sigcomm Computer Communication Review*. ACM: New York, USA, 2011; 266–277.

24. Al-Fares M, Loukissas A, Vahdat A. A scalable, commodity data center network architecture. In *ACM Sigcomm Computer Communication Review*, Vol. 38. ACM: New York, USA, 2008; 63–74.

25. Akhlaghi S, Kiani A, Ghanavati MR. Cost-bandwidth tradeoff in distributed storage systems. *Computer Communications* 2010; 33(17): 2105–2115.

26. Boyd S, Vandenberghe L. *Convex Optimization*. Cambridge university press: Cambridge, UK, 2009.

27. Yeung RW. *Information Theory and Network Coding*. Springer: Berlin, Germany, 2008.

28. Gerami M, Xiao M. Exact optimized-cost repair in multi-hop distributed storage networks. In *Proceedings of IEEE International Conference on Communication (ICC)*, Sydney, Australia, 2014; 4120–4124.

29. Ho T, Méard M, Koetter R, Karger DR, Effros M, Shi J, Leong B. A random linear network coding approach to multicast. *IEEE Transactions on Information Theory* 2006; 52(10): 4413–4430.

30. Jaggi S, Sanders P, Chou PA, Effros M, Egner S, Jain K, Tolhuizen LM. Polynomial time algorithms for multicast network code construction. *IEEE Transactions on Information Theory* 2005; 51(6): 1973–1982.

31. Papadimitriou CH, Steiglitz K. *Combinatorial Optimization: Algorithms and Complexity*. Courier Corporation: New York, USA, 1982.