Generalized Optimal Storage Scaling via Network Coding

Yuchong Hu[1,3], Xiaoyang Zhang[1], Patrick P. C. Lee[2], and Pan Zhou[1]
[1]Huazhong University of Science and Technology [2]The Chinese University of Hong Kong
[3]Shenzhen Huazhong University of Science and Technology Research Institute

***Abstract*—It is critical to support efficient scaling in distributed storage systems so as to meet increasing storage demands with new storage nodes. However, the scaling process incurs substantial scaling bandwidth due to reorganizing currently stored data to new storage nodes. Recent work has applied network coding to minimize scaling bandwidth for a special case where (n, k) MDS codes are scaled to (n', k') MDS codes for $n' - k' = n - k$. In this paper, we extend the results and prove the minimum scaling bandwidth for a more general setting where (n, k) MDS codes are scaled to (n', k') MDS codes for $n' > n$ and $k' \geq k$. Furthermore, we present a family of MDS code construction that achieves optimal scaling from (n, k) to (n', k') where $k = k'$.**

I. Introduction

Distributed storage systems [3], [4] adopt erasure coding to ensure fault-tolerant storage with low redundancy. Here, we consider a special class of erasure codes called (n, k) Maximum Distance Separable (MDS) codes, where $k < n$. In (n, k) MDS codes, we divide an original file of size M into k *blocks* of size M/k each, and encode them into n coded blocks also of size M/k each, such that any k out of n coded blocks can reconstruct the original file (called the *MDS property*).

To adapt to increasing storage demands, new nodes are often added to erasure-coded storage systems. This motivates us to study the *storage scaling* problem, in which the objective is to re-distribute all coded data across all nodes to balance the storage usage. Since storage scaling inevitably incurs substantial *scaling bandwidth* (i.e., the amount of traffic triggered during the scaling process) across the network, many studies (e.g., [5], [9], [10]) have proposed scaling approaches to mitigate the scaling bandwidth.

A recent study [11] presents the first work that applies network coding [1] to minimize the scaling bandwidth in erasure-coded storage by allowing storage nodes to send encoded data during scaling. However, it only addresses a special case that scales from (n, k) MDS codes to (n', k') MDS codes for $n' - k' = n - k$. The scaling analysis for more general cases remains unexplored.

In this paper, we consider a more general storage scaling setting from (n, k) MDS codes to (n', k') MDS codes for $n' > n$ and $k' \geq k$. We prove the information-theoretically minimum scaling bandwidth using the information flow graph model [1]. We further present a family of MDS code construction that achieves the minimum scaling bandwidth for scaling from (n, k) to (n', k') for $k = k'$, which covers a scaling scenario that keeps k intact while increasing the redundancy (by increasing n) for higher fault tolerance.

This work was supported by the National Natural Science Foundation of China (61502191, 61502190, 61602197, 61772222), Fundamental Research Funds for the Central Universities (2017KFYXJJ065, 2016YXMS085), Shenzhen knowledge innovation program (JCYJ20170307172447622), ZTE industry-academia-research cooperation funds and Research Grants Council of Hong Kong (GRF 14216316 and CRF C7036-15G).

II. Problem

We define the scaling problem from (n, k) MDS codes to (n', k') MDS codes, where (i) $n' > n$ and (ii) $k' \geq k$. Case (i) states that there are $n' - n$ new nodes added into the storage system, indicating that the scaling process deals with increasing storage demands. Case (ii) states that the capacity of each node before scaling (i.e., M/k) is no more than that of each node after scaling (i.e., M/k'), implying that the scaling process migrates data from the existing nodes to new nodes.

We perform storage scaling from (n, k) to (n', k') for a data file of size M in two steps. In the first step, each existing node X_i $(1 \leq i \leq n)$ encodes its stored data of size $\frac{M}{k}$ into some encoded data, deletes $\frac{M}{k} - \frac{M}{k'}$ of its stored data, and only stores data of size $\frac{M}{k'}$. In the second step, each new node $Y_{i'}$ $(1 \leq i' \leq n' - n)$ downloads the encoded data from each X_i $(1 \leq i \leq n)$ and encodes all its downloaded data into the stored data of size $\frac{M}{k'}$. Let β denote the bandwidth between any existing node X_i to any new node $Y_{i'}$; in other words, each $Y_{i'}$ downloads at most β units of encoded data from X_i.

Our goal is to minimize the scaling bandwidth, while preserving the MDS property; equivalently, we aim to minimize β, while the data file can be reconstructed from any k nodes.

III. Model

We construct an information flow graph $\mathcal{G}$ from (n, k) to (n', k'), as shown in Figure 1.

Nodes in $\mathcal{G}$:

- A virtual source S and a data collector T are added as the source and destination nodes of $\mathcal{G}$, respectively.
- Each existing storage node X_i $(1 \leq i \leq n)$ is represented by (i) an input node X_i^{in}, (ii) a middle node X_i^{mid}, (iii) an output node X_i^{out}, (iv) a directed edge $X_i^{in} \rightarrow X_i^{mid}$ with capacity $\frac{M}{k}$ (i.e., the amount of data stored in X_i before scaling), and (v) a directed edge $X_i^{mid} \rightarrow X_i^{out}$ with capacity $\frac{M}{k'}$ (i.e., the amount of data stored in X_i after scaling).
- Each new storage node $Y_{i'}$ $(1 \leq i' \leq n' - n)$ is represented by (i) an input node $Y_{i'}^{in}$, (ii) an output node $Y_{i'}^{out}$, and (iii) a directed edge $Y_{i'}^{in} \rightarrow Y_{i'}^{out}$ with capacity $\frac{M}{k'}$ (i.e., the amount of data stored in $Y_{i'}$).

Edges in $\mathcal{G}$:

- A directed edge $S \rightarrow X_i^{in}$ is added for every i $(1 \leq i \leq n)$ with an infinite capacity for data distribution.

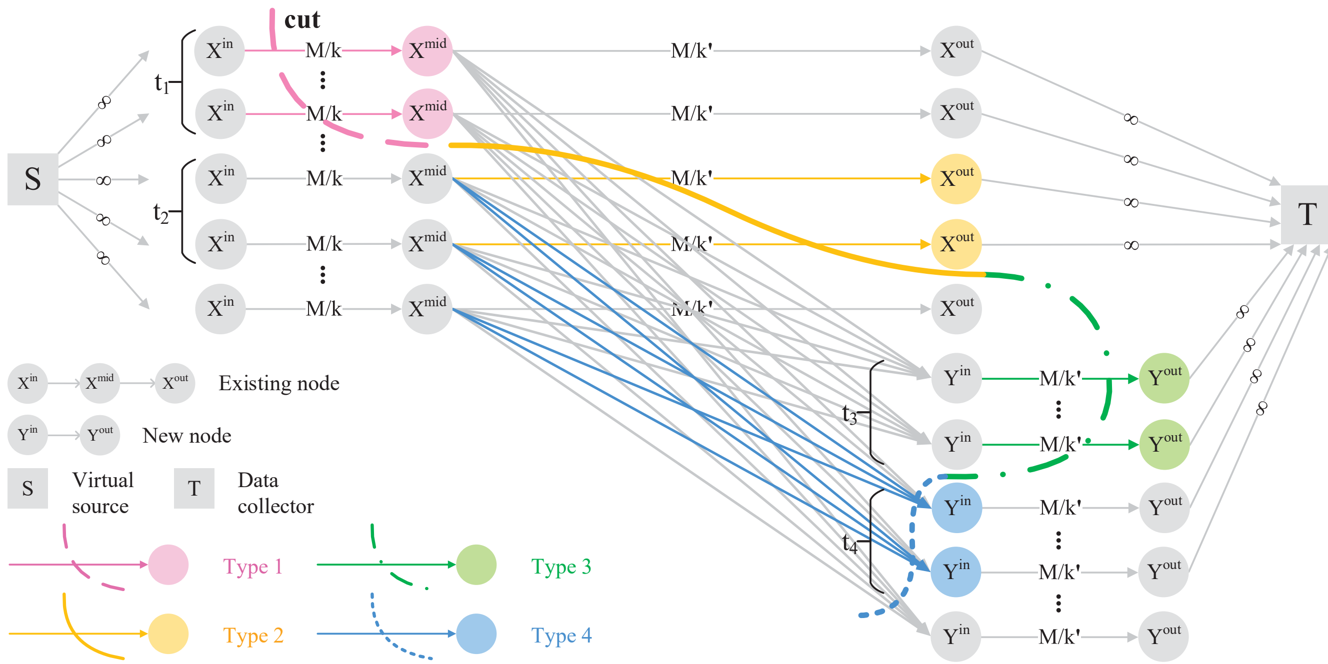

Fig. 1. Information flow graph $\mathcal{G}$ for scaling from (n, k) to (n', k'), where $n' > n$ and $k' \geq k$.

- A directed edge $X_i^{mid} \to Y_{i'}^{in}$ is added for every i ($1 \leq i \leq n$) and i' ($1 \leq i' \leq n' - n$) with capacity β.
- We select any k output nodes and a directed edge is added from each of them to T with an infinite capacity for data reconstruction.

To minimize β, we analyze the capacities of all possible min-cuts in $\mathcal{G}$. A *cut* is a collection of directed edges, such that any path from S to T must have at least one edge in the cut. A *min-cut* is the cut that has the minimum sum of capacities of all its edges. To preserve the MDS property after scaling, we need to consider $\binom{n'}{k'}$ possible data collectors at T. Thus, both the number of variants of $\mathcal{G}$ and the number of possible min-cuts are also $\binom{n'}{k'}$.

Let $(\mathcal{C}, \bar{\mathcal{C}})$ be some cut of $\mathcal{G}$, where $S \in \mathcal{C}$ and $T \in \bar{\mathcal{C}}$. Note that we do not consider the cuts with edges directed from S or to T, as such edges have an infinite capacity. For the remaining cuts, we characterize them by classifying the storage nodes into four types based on the nodes in $\bar{\mathcal{C}}$ (see Figure 1 for details):

- *Type 1*: Both X_i^{mid} and X_i^{out} are in $\bar{\mathcal{C}}$ for some $i \in [1, n]$;
- *Type 2*: X_i^{mid} is in $\mathcal{C}$, while X_i^{out} is in $\bar{\mathcal{C}}$, for some $i \in [1, n]$;
- *Type 3*: $Y_{i'}^{in}$ is in $\mathcal{C}$, while $Y_{i'}^{out}$ is in $\bar{\mathcal{C}}$, for some $i' \in [1, n' - n]$; and
- *Type 4*: Both $Y_{i'}^{in}$ and $Y_{i'}^{out}$ are in $\bar{\mathcal{C}}$ for some $i' \in [1, n' - n]$.

Suppose that T connects to t_i nodes of Type i ($1 \leq i \leq 4$). To make data reconstruction viable after scaling, we require:

$$t_1 + t_2 + t_3 + t_4 = k'. \tag{1}$$

IV. Analysis

We now derive the lower bound of β by analyzing the min-cuts of $\mathcal{G}$. Our analysis is similar to that of the classical repair problem via network coding [2]. Although both scaling and repair problems aim to minimize bandwidth, there exists one key difference: in scaling, if the data collector T selects some existing nodes (say one of them is X) and some new nodes (say one of them is Y), then there may be less than β units of *effective information* from X to Y. The reason is that X has offered all its information to T; even if it transfers β units to Y, some of them are not seen as effective information from the perspective of T. For example, Figure 2 depicts the scaling from $(3, 2)$ to $(4, 2)$. Although the bandwidths of all the scaling links between existing nodes and new nodes are defined as β, the effective information from X_3^{mid} to Y_1^{in} is actually zero. Since X_3^{out} is selected by T and will provide $M/2$ units to T, X_3^{mid} cannot transmit additional effective information to Y_1^{in}. It motivates us to define β^* as effective information out of β. For example, in Figure 2, the effective information from X_3^{mid} to Y_1^{in} is $\beta^* = 0$, and the capacity of the cut is actually $2\beta + M/2$.

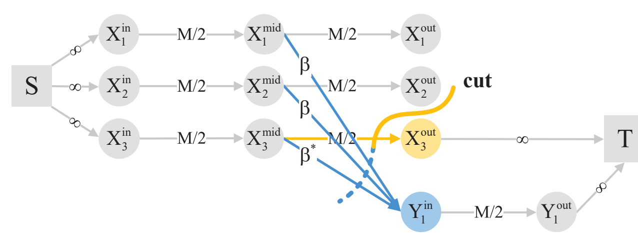

Fig. 2. Information flow graph $\mathcal{G}$ for scaling from $(3, 2)$ to $(4, 2)$.

Based on the definition of β^*, we see that β and β^* are subject to the following inequalities:

$$\begin{cases} \beta \geq \beta^*, \\ \beta^* \leq \frac{\frac{M}{k} - \frac{M}{k'}}{t_4}. \end{cases} \tag{2}$$

Here, $\beta^* \leq \frac{\frac{M}{k} - \frac{M}{k'}}{t_4}$ means that each existing node can transmit at most $\frac{M}{k} - \frac{M}{k'}$ units to the t_4 selected new nodes.

Let $\Lambda(t_1, t_2, t_3, t_4)$ denote the capacity of a cut. We derive Λ as follows:

- Each storage node of Type 1 contributes $\frac{M}{k}$ to Λ;
- Each storage node of Type 2 contributes $\frac{M}{k'}$ to Λ;
- Each storage node of Type 3 contributes $\frac{M}{k'}$ to Λ; and
- Each storage node of Type 4 contributes $(n-t_1-t_2)\beta+t_2\cdot\beta^*$ to Λ.

Thus, we have:

$$\begin{aligned}\Lambda = \quad & t_1 \cdot \tfrac{M}{k} + t_2 \cdot \tfrac{M}{k'} + t_3 \cdot \tfrac{M}{k'} + \\ & t_4 \cdot ((n-t_1-t_2)\beta + t_2 \cdot \beta^*). \end{aligned} \tag{3}$$

We consider three cases of Λ as follows.

A. Case 1: $k = k'$

When $k = k'$, based on the example in Figure 2, we have

$$\beta^* = 0. \tag{4}$$

Thus, Equation (3) can reduce to:

$$\Lambda = t_1 \cdot \frac{M}{k} + t_2 \cdot \frac{M}{k'} + t_3 \cdot \frac{M}{k'} + t_4 \cdot (n - t_1 - t_2)\beta. \tag{5}$$

In addition, due to Equation (4), Equation (2) is satisfied.

We first give a necessary condition of the lower bound of β by analyzing a specific case as follows:

$$\begin{aligned} t_1 &= 0; \\ t_2 &= k' - 1; \\ t_3 &= 0; \\ t_4 &= 1. \end{aligned}$$

In this case, Equation (5) can reduce to:

$$\Lambda = (k'-1) \cdot \frac{M}{k'} + (n - k' + 1) \cdot \beta. \tag{6}$$

Since the capacities of all possible min-cuts of $\mathcal{G}$ are at least M for valid file reconstruction, we have $\Lambda \geq M$. By Equation (6), we have

$$\beta \geq \frac{M}{(n-k'+1)k'}. \tag{7}$$

To show that the lower bound in Equation (7) is actually tight, we can analyze the capacities of all possible min-cuts of $\mathcal{G}$ of Case 1 via the following lemma.

Lemma 1. *Suppose that $k = k'$ and β is equal to its lower bound $\frac{M}{(n-k'+1)k'}$. Then the capacity of each possible min-cut of $\mathcal{G}$ is at least M.*

Proof: By Equation (7), Equation (5) can reduce to:

$$\Lambda \geq t_1 \cdot \frac{M}{k} + t_2 \cdot \frac{M}{k'} + t_3 \cdot \frac{M}{k'} + t_4 \cdot \frac{M \cdot (n - t_1 - t_2)}{(n-k'+1)k'}. \tag{8}$$

By Equation (1) and $k = k'$, Equation (8) can reduce to:

$$\Lambda \geq M + t_4 \cdot \frac{M}{k'} \cdot \frac{k' - 1 - t_1 - t_2}{n - k' + 1}. \tag{9}$$

When $t_4 = 0$, $\Lambda \geq M$. When $t_4 \geq 1$, by Equation (1), $k' - 1 - t_1 - t_2 \geq 0$ and hence $\Lambda \geq M$. The lemma holds. □

B. Case 2: $k < k'$ and $\frac{n}{k} \geq \frac{n'}{k'}$

Similar to Case 1, we first give a necessary condition of the lower bound of β and obtain β^*.

Clearly, each new storage node $Y_{i'}$ $(1 \leq i' \leq (n'-n))$ must receive at least $\frac{M}{k'}$ units of data from all existing storage nodes X_i's $(1 \leq i \leq n)$ over the links with capacity β each. Thus, we have

$$\beta \geq \frac{M}{nk'}. \tag{10}$$

Let β^* be equal to $\frac{M}{nk'}$. Then we need to show that β and β^* in Case 2 satisfy the conditions of Equation (2). Clearly, the first equation of Equation (2) is satisfied. The second equation of Equation (2) can reduce to

$$\frac{\frac{M}{k} - \frac{M}{k'}}{t_4} \geq \frac{M}{nk'}. \tag{11}$$

Type 4 only has new storage nodes, so $t_4 \leq n' - n$. Then

$$\frac{\frac{M}{k} - \frac{M}{k'}}{t_4} \geq \frac{\frac{M}{k} - \frac{M}{k'}}{n'-n}. \tag{12}$$

By Equation (12), Equation (11) holds if

$$\frac{\frac{M}{k} - \frac{M}{k'}}{n'-n} - \frac{M}{nk'} \geq 0. \tag{13}$$

Equation (13) can reduce to

$$M \cdot \frac{k' \cdot n - k \cdot n'}{(n'-n) \cdot kk'n} \geq 0. \tag{14}$$

Since $\frac{n}{k} \geq \frac{n'}{k'}$, Equation (14) holds, so Equation (11) holds. Thus, the second equation of Equation (2) is satisfied.

To show the lower bound in Equation (10) is actually tight, we analyze the capacities of all possible min-cuts of $\mathcal{G}$ of Case 2 via the following lemma.

Lemma 2. *Suppose that $k < k'$, $\frac{n}{k} \geq \frac{n'}{k'}$ and β is equal to its lower bound $\frac{M}{nk'}$. Then the capacity of each possible min-cut of $\mathcal{G}$ is at least M.*

Proof: Given that $\beta^* = \beta$, Equation (3) can reduce to:

$$\Lambda = t_1 \cdot \frac{M}{k} + t_2 \cdot \frac{M}{k'} + t_3 \cdot \frac{M}{k'} + t_4 \cdot (n - t_1)\beta. \tag{15}$$

By Equation (1), and $n' - n \geq t_4$ (Type 4 only has new storage nodes), Equation (15) can reduce to:

$$\Lambda \geq M + t_1 \cdot M \cdot \frac{n \cdot k' - k \cdot n'}{kk'n}. \tag{16}$$

Due to $\frac{n}{k} \geq \frac{n'}{k'}$, the right hand side of Equation (16) must be at least M. The lemma holds. □

C. Case 3: $k < k'$ and $\frac{n}{k} < \frac{n'}{k'}$

We divide Case 3 into two sub-cases.

1) $\frac{\frac{M}{k} - \frac{M}{k'}}{t_4} \geq \frac{M}{nk'}$*:* Similar to Case 2 in Section IV-B, let $\beta^* = \beta = \frac{M}{nk'}$. Then we can ensure that Equation (2) is met and we have all possible $\Lambda \geq M$.

2) $\frac{\frac{M}{k}-\frac{M}{k'}}{t_4} < \frac{M}{nk'}$*:* In this sub-case, note that if $\beta^* = \beta = \frac{M}{nk'}$, then Equation (2) cannot be met, so the lower bound of β should be larger than $\frac{M}{nk'}$, i.e., $\frac{M}{nk'} < \beta$.

Let $\beta^* = \frac{\frac{M}{k}-\frac{M}{k'}}{t_4}$. Due to $\frac{\frac{M}{k}-\frac{M}{k'}}{t_4} < \frac{M}{nk'}$ and $\frac{M}{nk'} < \beta$, Equation (2) is met. Also, Equation (3) can reduce to

$$\Lambda = (t_1+t_2)\cdot\frac{M}{k} + t_3\cdot\frac{M}{k'} + t_4\cdot(n-t_1-t_2)\beta. \quad (17)$$

Then we give a necessary condition of the lower bound of β via analyzing a special case in which $t_3 = 0$. Equation (17) can now reduce to:

$$\Lambda = (t_1+t_2)\cdot\frac{M}{k} + t_4\cdot(n-t_1-t_2)\beta. \quad (18)$$

Since the capacities of all the possible min-cuts of $\mathcal{G}$ are at least M for valid file reconstruction, we have $\Lambda \geq M$. Then by Equation (18), we have

$$\beta \geq \frac{M}{k}\cdot\frac{k-(t_1+t_2)}{(n-(t_1+t_2))(k'-(t_1+t_2))}. \quad (19)$$

To obtain the maximum value of the right hand side of Equation (19), we first determine the range of (t_1+t_2).

Due to $\frac{\frac{M}{k}-\frac{M}{k'}}{t_4} < \frac{M}{nk'}$, Equation (1) can reduce to

$$\frac{\frac{M}{k}-\frac{M}{k'}}{k'-(t_1+t_2+t_3)} < \frac{M}{nk'}. \quad (20)$$

Since $t_3 = 0$, Equation (20) can reduce to:

$$t_1+t_2 < \frac{kk'+n(k-k')}{k}. \quad (21)$$

Thus, we have

$$(t_1+t_2)_{max} = \begin{cases} \frac{kk'+n(k-k')}{k}-1, & \frac{nk'}{k} \text{ is integral,} \\ \lfloor\frac{kk'+n(k-k')}{k}\rfloor, & \frac{nk'}{k} \text{ is decimal.} \end{cases} \quad (22)$$

By Equation (1) and due to $t_3+t_4 \leq n'-n$ (nodes of Type 3 and Type 4 are all new storage nodes), we have $k'-(n'-n) \leq t_1+t_2$. Thus, we have

$$(t_1+t_2)_{min} = \begin{cases} k'-(n'-n), & k' \geq (n'-n), \\ 0, & k' < (n'-n). \end{cases} \quad (23)$$

Based on the right-hand side of Equation (19), we define a function $f(t_1+t_2)$ as follows:

$$f(t_1+t_2) = \frac{M}{k}\cdot\frac{k-(t_1+t_2)}{(n-(t_1+t_2))(k'-(t_1+t_2))}. \quad (24)$$

Through the derivation of Equation (24), we work out the maximum of the right hand side of Equation (19) as follows:

$$\begin{cases} f((t_1+t_2)_{max}), & (t_1+t_2)_{max} \leq Z, \\ \max(f(\lceil Z\rceil), f(\lfloor Z\rfloor)), & (t_1+t_2)_{min} \leq Z \leq (t_1+t_2)_{max}, \\ f((t_1+t_2)_{min}), & Z \leq (t_1+t_2)_{min}. \end{cases} \quad (25)$$

where $\quad Z = k - \sqrt{(n-k)(k'-k)}$.

To show the lower bound in Equation (25) is actually tight, we analyze the capacities of all possible min-cuts of $\mathcal{G}$ of Case 3 under the condition $\frac{\frac{M}{k}-\frac{M}{k'}}{t_4} < \frac{M}{nk'}$.

Lemma 3. *Suppose that $k < k'$, $\frac{n}{k} < \frac{n'}{k'}$, and β is equal to its lower bound given by Equation (25). Then the capacity of each possible min-cut of $\mathcal{G}$ is at least M.*

Proof: Since β is equal to its lower bound given by Equation (25), Equation (19) holds. By Equation (19), Equation (17) can reduce to:

$$\Lambda \geq (t_1+t_2)\cdot\frac{M}{k} + t_3\cdot\frac{M}{k'} + t_4\cdot\frac{M}{k}\cdot\frac{k-(t_1+t_2)}{k'-(t_1+t_2)}. \quad (26)$$

By Equation (1), Equation (26) can reduce to:

$$\Lambda \geq M + M(t_1+t_2)(k'-k)\cdot\frac{k'-(t_1+t_2)-t_4}{kk'(k'-(t_1+t_2))}. \quad (27)$$

Since $k' \geq t_1+t_2+t_4$ (see Equation (1)), the right hand side of Equation (27) must be at least M. The lemma holds. □

Lemma 4 ([2])**.** *If the capacity of each possible min-cut of $\mathcal{G}$ is at least the original file size M, there exists a random linear network coding scheme guaranteeing that T can reconstruct the original file for any connection choice, with a probability that can be driven arbitrarily high by increasing the field size.*

Theorem 1. *For scaling from (n,k) to (n',k'), the bounds derived from Lemmas 1, 2, and 3 are tight.*

Proof: The existence of random linear codes based on Lemma 4 makes the derived bounds tight. □

V. Code Construction for $k = k'$

Theorem 1 and Lemma 1 provide the tight lower bound of β when $k = k'$. In this section, we show how to construct a family of random linear codes, such that the scaling is optimal by satisfying $\beta = \frac{M}{(n-k+1)k}$ (i.e., $\frac{M}{(n-k'+1)k'}$ when $k = k'$) while maintaining the MDS property after scaling.

To explain our construction, we extend our system model in Section III. We first split the file of size M evenly into qk original blocks where $q = n-k+1$, and encode them into qn coded blocks. We distribute them into n existing nodes $X_1, X_2, \cdots, X_n$, each of which stores q coded blocks. The (n,k) MDS property is satisfied, i.e., the qk coded blocks of any k out of n nodes can reconstruct the qk original blocks. Here, each coded block has size equal to the lower bound of $\beta = \frac{M}{(n-k+1)k}$.

For the j^{th} coded blocks on the i^{th} node (where $1 \leq i \leq n$ and $1 \leq j \leq q$), it is formed by a linear combination of the qk original blocks over a finite field $\mathbb{F}$. Thus, we let $\mathbf{p}_{i,j}$ be a column vector of size qk specifying the coefficients for the above linear combination, and also let $\mathbf{P}_i$ be a $qk \times q$ matrix comprising the column vectors $\{\mathbf{p}_{i,j}\}_{1\leq j\leq q}$. Clearly, the original file can be reconstructed by decoding qk coded blocks of any k nodes via inverting an encoding matrix [7]. Now we can specify our code construction in the way that uses $\mathbf{P}_i$ and $\mathbf{p}_{i,j}$ to refer to the all the q blocks and the j^{th} block stored in X_i, respectively.

The scaling from (n,k) to (n',k') works as follows. Due to $k' = k$, each new node $Y_{i'}$ also has q blocks. During scaling, each existing node X_i (where $1 \leq i \leq n$) encodes all its blocks into $n' - n$ new blocks, each of which is defined as $\mathbf{P}_i \cdot \mathbf{c}_{i,i'}$, where $\mathbf{c}_{i,i'}$ denotes a coefficient vector of size q, (where $1 \leq i' \leq n' - n$), and then transmits the $n' - n$ new blocks to $Y_1, \ldots, Y_{n'-n}$ in order. In this way, each new node $Y_{i'}$ (where $1 \leq i' \leq n' - n$) receives n new blocks, and then encodes all the n received blocks into q coded blocks denoted by

$$\mathbf{P}'_{i'} = [\mathbf{P}_1 \cdot \mathbf{c}_{1,i'}, \cdots, \mathbf{P}_n \cdot \mathbf{c}_{n,i'}] \cdot \mathbf{D}_{i'}, \tag{28}$$

where $\mathbf{D}_{i'}$ is a $n \times q$ coefficient matrix, and $1 \leq i' \leq n' - n$.

Suppose that the MDS property is satisfied before scaling. To maintain the MDS property after scaling, we ensure that for any k (i.e., k') nodes collected by T, the collection composed of the qk vectors of these collected k nodes, denoted by $\mathbf{W}$, has full rank. Let T be connected with u nodes from the existing nodes and v nodes from the new nodes, satisfying that $u + v = k$. When $v = 0$, it is clear that the MDS property is satisfied after scaling, so we only consider $v \geq 1$. By $u+v = k$ and $q = n - k + 1$, we can have

$$(n - u)v \geq qv. \tag{29}$$

We now construct the codes that maintain the MDS property for scaling from (n, k) to (n', k') where $k = k'$.

Theorem 2. *If we divide the original file of size M into qk blocks where $q = n - k + 1$, then there exists a linear coding construction defined in the finite field $\mathbb{F}$ for the optimal scaling from (n, k) to (n', k') where $n < n'$ and $k = k'$, such that the MDS property is still maintained with a probability arbitrarily driven to 1 by increasing the field size of $\mathbb{F}$.*

Proof: Suppose that before scaling $\{\mathbf{P}_i\}_{1\leq i\leq n}$ satisfies the MDS property initially. We show that there exist assignments of $\mathbf{c}_{i,i'}$ and $\mathbf{D}_{i'}$, such that $\mathbf{W} = \{\mathbf{P}_1; \cdots; \mathbf{P}_u; \mathbf{P}'_1; \cdots; \mathbf{P}'_v\}$ has full rank.

By Equation (28), we have $\mathbf{W} = \{\mathbf{P}_1, \cdots, \mathbf{P}_u;$

$$[\mathbf{P}_1 \cdot \mathbf{c}_{1,1}, \cdots, \mathbf{P}_n \cdot \mathbf{c}_{n,1}] \cdot \mathbf{D}_1, \cdots,$$
$$[\mathbf{P}_1 \cdot \mathbf{c}_{1,v}, \cdots, \mathbf{P}_n \cdot \mathbf{c}_{n,v}] \cdot \mathbf{D}_v\}.$$

Clearly, $span(\mathbf{W}) = \{\mathbf{P}_1; \cdots; \mathbf{P}_u;$

$$[\mathbf{P}_{u+1} \cdot \mathbf{c}_{u+1,1}; \cdots; \mathbf{P}_n \cdot \mathbf{c}_{n,1}] \cdot \mathbf{D}_1^{n-u}, \cdots;$$
$$[\mathbf{P}_{u+1} \cdot \mathbf{c}_{u+1,v}; \cdots; \mathbf{P}_n \cdot \mathbf{c}_{n,v}] \cdot \mathbf{D}_v^{n-u}\}, \tag{30}$$

where $\mathbf{D}_{i'}^{n-u}$ is a matrix composed of the last $n - u$ row vectors of $\mathbf{D}_{i'}$.

By Equations (29) and (30), we can tune $\mathbf{c}_{i,i'}$ and $\mathbf{D}_{i'}^{n-u}$ ($1 \leq i \leq n$ and $1 \leq i' \leq n' - n$) such that the collection $\{[\mathbf{P}_{u+1} \cdot \mathbf{c}_{u+1,l}; \cdots; \mathbf{P}_n \cdot \mathbf{c}_{n,l}] \cdot \mathbf{D}_l^{n-u}, 1 \leq l \leq v\}$ is composed of vq vectors of v nodes out of $\mathbf{P}_{u+1}, \ldots, \mathbf{P}_n$. Since $\{\mathbf{P}_i\}_{1\leq i\leq n}$ satisfies the MDS property initially, the span of $\{\mathbf{P}_1; \cdots; \mathbf{P}_u\}$ plus $\{[\mathbf{P}_{u+1}\cdot\mathbf{c}_{u+1,l}; \cdots; \mathbf{P}_n\cdot\mathbf{c}_{n,l}]\cdot\mathbf{D}_l^{n-u}, 1 \leq l \leq v\}$ have rank $uq + vq$. Since $u + v = k$, $span(\mathbf{W})$ has full rank.

We can show that $\det(\mathbf{W})$ is a nonzero number for a certain assignment of $\mathbf{c}_{i,i'}$ and $\mathbf{D}_{i'}^{n-u}$ because $span(\mathbf{W})$ has full rank. This means that $\det(\mathbf{W})$ is a non-zero polynomial. Thus, $\det(\mathbf{W}) \neq 0$ holds with a probability arbitrarily driven to one by increasing the field size of $\mathbb{F}$, as a result of the Schwartz-Zippel Theorem [6]. Thus, Theorem 2 concludes. □

VI. Related Work

Many prior studies propose to mitigate the scaling bandwidth, e.g., FastScale [12], GSR [9]. However, these studies address storage scaling in RAID arrays. Some follow-up studies consider cases in distributed environments. For example, Rai et al. [8] propose a coding scheme that can switch between two given different (n, k) settings. Huang et al. [5] reduce the scaling bandwidth in erasure-coded distributed storage systems. Zhang et al. [11] apply network coding to storage scaling to minimize the scaling bandwidth, yet they only consider special cases when scaling from (n, k) to (n', k') for $n' - k' = n - k$. This paper generalizes the scaling cases in [11] and present formal analysis on the optimal storage scaling.

VII. Conclusions

We study generalized storage scaling via network coding to handle increasing storage demands, and present two key findings. First, we prove, via the information flow graph model, the minimum scaling bandwidth when (n, k) MDS codes are scaled to (n', k') MDS codes for $n' > n$ and $k' \geq k$. Also, we construct a family of MDS codes that achieves minimum scaling bandwidth when scaling (n, k) to (n', k') for $k = k'$. Our future work is to address the scale-down case for $n > n'$.

References

[1] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung. Network Information Flow. *IEEE Trans. on Info. Theory*, 46(4):1204–1216, Jul 2000.

[2] A. G. Dimakis, P. B. Godfrey, Y. Wu, M. Wainwright, and K. Ramchandran. Network Coding for Distributed Storage Systems. *IEEE Trans. on Info. Theory*, 56(9):4539–4551, Sep 2010.

[3] D. Ford, F. Labelle, F. I. Popovici, M. Stokely, V.-A. Truong, L. Barroso, C. Grimes, and S. Quinlan. Availability in Globally Distributed Storage Systems. In *Proc. of USENIX OSDI*, 2010.

[4] C. Huang, H. Simitci, Y. Xu, A. Ogus, B. Calder, P. Gopalan, J. Li, and S. Yekhanin. Erasure Coding in Windows Azure Storage. In *Proc. of USENIX ATC*, 2012.

[5] J. Huang, X. Liang, X. Qin, P. Xie, and C. Xie. Scale-RS: An Efficient Scaling Scheme for RS-coded Storage Clusters. *IEEE Trans. on Parallel and Distributed Systems*, 26(6):1704–1717, 2015.

[6] R. Motwani and P. Raghavan. Randomized Algorithms. In *Cambridge University Press*, 1995.

[7] J. S. Plank. A Tutorial on Reed-Solomon Coding for Fault-Tolerance in RAID-like Systems. *Software - Practice & Experience*, 27(9):995–1012, Sep 1997.

[8] B. K. Rai, V. Dhoorjati, L. Saini, and A. K. Jha. On Adaptive Distributed Storage Systems. In *Proc. of IEEE ISIT*, 2015.

[9] C. Wu and X. He. GSR: A Global Stripe-based Redistribution Approach to Accelerate RAID-5 Scaling. In *Proc. of IEEE ICPP*, 2012.

[10] S. Wu, Y. Xu, Y. Li, and Z. Yang. I/O-Efficient Scaling Schemes for Distributed Storage Systems with CRS Codes. *IEEE Trans. on Parallel and Distributed Systems*, 27(9):2639–2652, Sep 2016.

[11] X. Zhang, Y. Hu, P. P. C. Lee, and P. Zhou. Toward Optimal Storage Scaling via Network Coding: From Theory to Practice. In *Proc. of IEEE INFOCOM*, 2018.

[12] W. Zheng and G. Zhang. FastScale: Accelerate RAID Scaling by Minimizing Data Migration. In *Proc. of USENIX FAST*, 2011.