Capacity of dynamical storage systems

Ohad Elishco Alexander Barg

***Abstract*—We define a time-dependent model of erasure coding for distributed storage and estimate the average capacity of the network in the simple case of fixed link bandwidth that takes one of two given values. We show that if k data blocks are encoded into n blocks placed on n nodes of which n_1 have links with bandwidth greater than the remaining $n-n_1$ nodes by γ symbols, then the average capacity increases by $\Omega(\gamma(k-n_1)^2)$ symbols compared to the static model.**

I. Introduction

The problem of node repair based on erasure coding for distributed storage aims at optimizing the tradeoff of network traffic and storage overhead. In this form it was established by [1] from the perspective of network coding. This model was generalized in various ways such as concurrent failure of several nodes [2], heterogeneous architecture [3], [4], cooperative repair [5], and others. The existing body of works focuses on the failure of a node (or several nodes) and the ensuing reconstruction process, but puts less emphasis on the time evolution of the entire network and the inherent stochastic nature of the node failures. The static point of view of the system and of node repair leads to schemes based on the worst-case scenario in the sense that the amount of data to be stored is known in advance, and the repair capacity is determined by the least advantageous state of the network. Switching to evolving networks makes it possible to define and study the average amount of data moved through the network to accomplish repair, or, equivalently, determining the average file size that can be retrieved from the network once the capacities of the links from each of the nodes to the data collector are fixed to some values.

In this work we make first steps toward defining a dynamic model of the network with random failures. We observe that no gain is obtained if the links from each of the nodes have the same capacity. Stepping away from this assumption, we adopt a simple model of heterogeneous storage wherein the network is formed of two disjoint groups of nodes with unequal communication costs, which was proposed in the static case in [3]. We show that, under the assumption of uniform failure rates of the nodes, it is possible to increase the *average* size of the file stored in the system.

In Section II we present the dynamical model and basic definitions. In Section III we define the fixed-cost model and prove a lower bound on the average capacity, which forms the main result of this paper. The evolution of the network's information flow graph is formalized as a Markov chain similar to those occurring in card shuffling problems, and we make use of the classic results about their mixing times.

The authors are with ISR/Dept. of ECE, University of Maryland, College Park, MD 20817, USA, emails ohadeli@umd.edu and abarg@umd.edu. A. Barg is also with Inst. Inform. Trans. Probl., Russian Academy of Sciences, Moscow, Russia.

Research supported by NSF grants CCF1618603 and CCF1814487.

II. Model Definition

In this section we define a storage network that evolves in time and describe the basic assumptions that characterize this evolution. We also define a sequence of information flow graphs, which enables us to define capacity of a randomly evolving network.

A. Evolution of the network. A storage network is a triple $\mathcal{N} = (V, DC, CU)$ where $V = (v_1, \dots, v_n)$ is a set of n nodes (storage units), DC is a node called data collector, and CU is a centralized computing unit (repair center). Every node v_i, $i \in [n] \triangleq \{1, 2, \dots, n\}$, has the ability to store up to α symbols over some finite field F. To store a set of M symbols (a "file"), we divide it into k information blocks viewed as vectors over F which are encoded with an (n, k) vector code $\mathcal{C}$. The coordinates of the codeword are vectors over F, and each coordinate is stored in its own node in V. To read the file, the DC accesses at least k nodes, obtaining the information stored in them, and attempts to retrieve the file.

The storage network evolves in time, which we assume to be discrete. At time $t = 0$, the encoded file is stored in the network. The time units $t = 1, 2, \dots$ indicate node failures (only those times will be taken into account in our model). Let $s = (s_1, s_2, \dots) \in V^\infty$ be the sequence of failed nodes, where s_t is the node that fails at time t. When a failure occurs, the CU corrects it (replacing the erased information so that message retrieval still be possible) by downloading information from d other (helper) nodes, finding the value of the failed node and creating a new node to replace it. In this work we assume that $d = n - 1$, i.e., the CU collects information from all of the other nodes to perform the repair.

Given $\mathcal{N}$ and the sequence s, we define a sequence of directed weighted graphs $\mathcal{X}_t, t \geqslant 0$, called *information flow graphs*, where $\mathcal{X}_t$ is a subgraph of $\mathcal{X}_{t+1}$ for each t.

1: Let $V_0 = V \cup \tilde{v}$, i.e., all the nodes in $\mathcal{N}$ and a *source node* node $\tilde{v}$, and define the graph $\mathcal{X}_0 = (V_0, E_0)$ with edges

$$E_0 = \{(\tilde{v}, v_i) \ : \ i \in [n]\}$$

where each edge has weight α. We will call the nodes in the set $\mathcal{A}_0 := V$ *active nodes* of the graph $\mathcal{X}_0$.

2: Suppose that $s_1 = v_{i_1}, i_1 \in [n]$ and define the new node $v_{i_1}^1$. The graph $\mathcal{X}_1 = (V_1, E_1)$ is formed as follows:

$$V_1 = V_0 \cup \{CU_1, v_{i_1}^1\}$$
$$E_1 = E_0 \cup \{(v_j, CU_1), j \in [n] \backslash \{i_1\}\} \cup (CU_1, v_{i_1}^1).$$

The set of active nodes of $\mathcal{X}_1$ is defined as $\mathcal{A}_1 := (\mathcal{A}_0 \backslash \{v_{i_1}\}) \cup \{v_{i_1}^1\}$.

3: Suppose we are given the graph $\mathcal{X}_{t-1}$. Suppose that $s_t = v_{i_t}^{t'}$ for some $t' < t$ is the value of the failed node ($s_t \in \mathcal{A}_{t-1}$ and $i_t \in [n]$). Define $\mathcal{X}_t(V_t, E_t)$ as follows:

$$V_t = V_{t-1} \cup \{CU_t, v_{i_t}^t\}$$
$$E_t = E_{t-1} \cup \{(u, CU_t) : u \in \mathcal{A}_{t-1} \backslash \{v_{i_t}^{t'}\}\} \cup (CU_t, v_{i_t}^t).$$

The set of active nodes of $\mathcal{X}_t$ is defined as $\mathcal{A}_t = (\mathcal{A}_{t-1} \backslash \{v_{i_t}^{t'}\}) \cup v_{i_t}^t$.

For any $t \geqslant 1$ the weight of the edge $(u, CU_t) = \beta_i$, where u corresponds to $v_i \in V$, and the weight of $(CU_t, v_{i_t}^t) = \alpha$.

The graph $\mathcal{X}_t$ captures the connectivity structure and the state of the network up to the time instant t. The weight of an edge β_i denotes the target average number of symbols transmitted through the corresponding link over time.

B. Data retrieval and network capacity. Information exchange in the network is performed over the links between the nodes. The Data Collector DC initiates the data retrieval by contacting at least k nodes in the set V. In the information flow graph $\mathcal{X}_t$, this process amounts to introducing a new node, DC_t, and connecting it to a subset of active nodes $S_t \subseteq \mathcal{A}_t$, $|S_t| \geqslant k$. The links from S_t to DC_t are assumed to have infinite capacity.

Let $\mathcal{N}$ be a storage network with the corresponding information flow graphs $\mathcal{X}_t, t \in \mathbb{N}$ and let s be a sequence of failed nodes. Denote by $C_t(S_t)$ the capacity at time t for S_t. Informally, $C_t(S_t)$ is the maximum file size that can be retrieved by DC_t at time t if it contacts the set S_t. Formally, $C_t(S_t)$ equals the edge weight of a minimum cut in $\mathcal{X}_t$ between $\left(\bigcup_{i=-1}^{t} \mathcal{A}_i \setminus \mathcal{A}_t\right)$ and DC_t, where we define $\mathcal{A}_{-1} = \tilde{v}$. Further, let C_t be the capacity at time t, i.e.,

$$C_t = C_t(\mathcal{A}_t) \triangleq \min_{S_t \subseteq \mathcal{A}_t,\, |S_t| \geqslant k} \{C_t(S_t)\}.$$

In this definition we assume that DC is not aware of the state of the network, and the minimum accounts for the worst case.

Definition 1 *Let $\mathcal{N}$ be a storage network and let s be a sequence of failed nodes. Define the* network capacity *as*

$$\mathsf{cap}(\mathcal{N}) \triangleq \limsup_{t\to\infty} \frac{1}{t} \sum_{i=1}^{t} C_i.$$

The main assumption that we make is that at any given time t, the failed node is chosen uniformly and independently from the set of nodes V, and thus s is a *sequence of RVs* which we henceforth denote by $\boldsymbol{S}$. This makes the C_t's and $\mathsf{cap}(\mathcal{N})$ into random variables. In this work we analyze the expected value of the capacity.

Definition 2 *Let $\mathcal{N}$ be a storage network and let $\boldsymbol{S}$ be a random sequence of failed nodes. The expected capacity is defined as*

$$\overline{\mathsf{cap}}(\mathcal{N}) \triangleq \mathbb{E}\left[\mathsf{cap}(\mathcal{N})\right].$$

It is clear that if $\beta_i = \beta$ for every $i \in [n]$, the fact that $\boldsymbol{S}$ is random will not affect the capacity which implies that the $\mathsf{cap}(\mathcal{N})$ is equal to the minimum cut. Hence, both $\mathsf{cap}(\mathcal{N})$ and its expectation in the case $\beta_i = \beta$ are given in [1].

At the same time, in many situations the links between the nodes have different capacities, and below we study storage networks with different β_is.

Each node in $\mathcal{A}_t$ is given by $v_i^{t_i}$ for some $t_i \leqslant t$ and may be identified with the node $v_i \in V$. Therefore, if at some point t_0 all the nodes have failed at least once, then for $t > t_0$ the order in which the nodes in $\mathcal{A}_t$ have failed may be identified with a permutation over the set V. For example, if $\mathcal{A}_t = \left\{v_1^{t_1}, v_2^{t_2}, \ldots, v_n^{t_n}\right\}$ with $t_i > 0$ for all i, then the corresponding permutation π_t is such that $\pi_t^{-1}(v_i) \leqslant \pi_t^{-1}(v_j)$ iff $v_i^{t_i}, v_j^{t_j} \in \mathcal{A}_t$ with $t_i \leqslant t_j$. We summarize this observation by saying that the values of C_t are parametrized by permutations of V (the permutations are well defined because $\mathcal{A}_t, t \geqslant t_0$ does not contain two nodes $v_i^t, v_i^{t'}$ that correspond to the same node in V).

Below we identify V and $[n]$ and consider $\pi_t, t \geqslant t_0$ as a permutation of either of these sets as appropriate. It is possible to obtain $\pi_t, t \geqslant t_0$ from $\boldsymbol{S}_1^t := \{\boldsymbol{S}_1, \ldots, \boldsymbol{S}_t\}$ by considering only the last appearance of each node.

Example 1 *Assume that $|V| = 5$ and assume that $\boldsymbol{S} = (v_1, v_2, v_3, v_4, v_5, v_2, v_1, v_5, \ldots)$. Then $\pi_1, \ldots, \pi_4$ are not defined and $\pi_5 = (v_1, v_2, v_3, v_4, v_5)$ since all the nodes had failed in $t = 5$. In $t = 6$ the node v_2 fails again, hence the new order is given by $\pi_6 = (v_1, v_3, v_4, v_5, v_2)$. This is because the second node appears twice in $\boldsymbol{S}_1^6$ and we consider only the last appearance. Following the same reasoning, $\pi_7 = (v_3, v_4, v_5, v_2, v_1)$ and $\pi_8 = (v_3, v_4, v_2, v_1, v_5)$.*

Since $\mathcal{A}_t$ is a function of $\boldsymbol{S}_1^t$, for $t \geqslant t_0$ the permutation π_t itself is random, and is a function of $\boldsymbol{S}_1^t$. Since for $t > t_0$ we may identify π_t and $\mathcal{A}_t$, we sometimes use the notation $C_t(\pi_t)$ and $C(\pi_t)$.

Lemma 1 *Let $\boldsymbol{S} = (\boldsymbol{S}_i, i \geqslant 1)$ be a sequence of independent RVs uniformly distributed on $[n]$. Then*

$$\overline{\mathsf{cap}}(\mathcal{N}) \overset{\text{a.s.}}{=} \lim_{t\to\infty} \frac{1}{t} \sum_{i=1}^{t} \mathbb{E}[C_i].$$

Proof: Let t_0 be the first time instance when all the vertices have failed at least once. Note that t_0 is a stopping time and each failed node is chosen uniformly and independently. Referring to the Coupon collector's problem [6, p.210], we obtain $\Pr(t_0 \geqslant cn \log n) \leqslant n^{1-c}, c \geqslant 1$. Thus, t_0 is finite almost surely.

By symmetry, π_{t_0} is distributed uniformly on the set of all permutations. Moreover, since $\boldsymbol{S}_i$ is chosen uniformly and independently, for $t \geqslant t_0$ we have that $\Pr(\pi_t = \pi | \pi_0^{t-1}) = \Pr(\pi_t = \pi | \pi_{t-1})$, so π_t is a Markov chain, which is irreducible and aperiodic. Because of this, a limiting distribution μ exists, and is unique and positive. Hence, as t grows, $\Pr(\pi_t) \to \mu(\pi_t)$. Together with the fact that C_t is uniformly bounded from above for all t, we obtain that the limit $\lim_{n\to\infty} \frac{1}{n} \sum_{t=1}^{n} \mathbb{E}[C_t]$ exists.

Now define $X_t = \frac{1}{t}\sum_{i=1}^{t} C_i$ and note that X_t is a function of $\boldsymbol{S}$. Following the previous discussion, for almost every $\boldsymbol{S}$, the sequence X_t converges. Since X_t is non-negative and upper bounded for every t, by the dominated convergence theorem we have $\lim_{t\to\infty}\mathbb{E}[X_t] = \mathbb{E}[\lim_{t\to\infty} X_t]$ (the limit exists a.s.), which is the desired result. ∎

The proof of Lemma 1 relies on the stopping time t_0 since the permutation π_t is not defined for $t < t_0$. Since t_0 is almost surely finite and since π_t is a Markov chain with limiting distribution, defining a starting permutation $\pi_0 = \mathrm{id}$ will not affect the expected capacity. Hence, from now on we assume $\pi_0 = \mathrm{id}$. Our problem is similar to the *Top in at random shuffle* mixing time, and we use the following result from [7, Thm.1].

Theorem 1 (ALDOUS AND DIACONIS) *Consider a deck of n cards. At time $t = 1, 2, \ldots$ take the top card and insert it in the deck at a random position. Let Q_t denote the distribution after t such shuffles and let U be the uniform distribution on the set of all permutations $\mathscr{S}_n$. Then for all $c \geqslant 0$ and $n \geqslant 2$, the total variation distance satisfies*

$$\|Q_{n\log n + cn} - U\|_{TV} \leqslant e^{-c}. \tag{1}$$

To connect this result to our problem, we note that random choice of the next failed node corresponds to selecting a random card from the deck and putting in at the bottom. The mixing time of this chain is stochastically equivalent to the mixing time of the *Top in at random shuffle*, and we obtain the following lemma.

Lemma 2 *Let $\mathcal{N}$ be a storage network with $|V| = n \geqslant 2$ nodes and let $\boldsymbol{S}$ be a random sequence of failed nodes. Consider the corresponding sequence of permutations $(\pi_t, t \geqslant 0)$ where $\pi_0 = \mathrm{id}$. Then for any $c \geqslant 0$, $n \geqslant 2$ and $\pi \in \mathscr{S}_n$,*

$$|\Pr(\pi_{n\log n + cn} = \pi) - \frac{1}{n!}| \leqslant e^{-c}.$$

Proof: Let $T \geqslant 1$ be a value of the time. Consider the time-reversed sequence $\tilde{\pi}_t = \pi_{T-t}, t \leqslant T$. The evolution of the sequence $\tilde{\pi}_t$ is described as follows: for any t take the last symbol $\pi_t(n)$ and insert it randomly in the middle. Observe that $\Pr(\pi_T = \pi) = \Pr(\tilde{\pi}_T = \mathrm{id}|\tilde{\pi_1} = \pi)$. Now use (1) and the definition of $\|\cdot\|_{TV}$ to claim that for $T = n\log n + cn, c \geqslant 0$, $\left|\Pr(\tilde{\pi}_T = \mathrm{id}|\tilde{\pi_0} = \pi) - \frac{1}{n!}\right| \leqslant e^{-c}$.

III. THE FIXED COST MODEL

In this section we define a fixed-cost model and present a lower bound on its expected capacity. Suppose that the set of nodes has the form $V = U \cup L$, where $U = (v_1, \ldots, v_{n_1})$ and $L = (v_{n_1+1} \ldots, v_{n_1+n_2})$ are disjoint non-empty subsets. Let $\beta_1 \geqslant \beta_2$ be such that

$$\beta_i = \begin{cases} \beta_1 & \text{if } v_i \in U \\ \beta_2 & \text{if } v_i \in L \end{cases},$$

This model is a dynamical equivalent of the model presented in [3].

Let $a \triangleq k - n_1$. We have the following lemma.

Lemma 3 *Let $\mathcal{N}$ be a fixed-cost storage network and let*

$$C \triangleq \min_{\pi\in\mathscr{S}_n}\{C(\pi)\} = \min_{\substack{t\geqslant 0,\\ \boldsymbol{S}\in V^\infty}}\{C_t\}.$$

Then

$$C = \sum_{i=1}^{\min\{n_1-1,k-1\}} \min\{n_1\beta_1 + n_2\beta_2 - i\beta_1, \alpha\} + \mathbb{1}_{\{k\geqslant n_1\}} \sum_{j=0}^{\max\{0,a\}} \min\{n_2\beta_2 - j\beta_2, \alpha\}. \tag{2}$$

The proof can be obtained from Lemma 4 below, and will be omitted.

We assume throughout that $a > 0$ because otherwise the file reconstruction problem is trivially solved by contacting k nodes in U. Expression (2) gives a lower bound for the cut C_t for all t and $\boldsymbol{S}$ in our storage network model. We note that by the definition of C, (2) also gives the value of the minimum cut for the static models of [3], [1].

In the next lemma we show that for $t \geqslant t_0$ the minimum cut will be obtained when DC_t chooses a set of k nodes that contains all nodes of U and some nodes from L.

Lemma 4 *If $\alpha \geqslant (n_1-1)\beta_1 + n_2\beta_2$ then C_t is obtained when DC_t selects k nodes which include the set U.*

The proof is set in terms of a dynamic programming problem, and is given in the Appendix.

Remark 1 *The proof of Lemma 4 also implies that, once we have chosen all the n_1 nodes from U and there are a more nodes to select, the optimum is obtained by choosing the a most recently failed nodes from L.*

Using Lemma 3, in the next example we show that the average total bandwidth in the dynamical model can be made lower than in the static case.

Example 2 Let $n = 10, k = 7$, $U = (v_1, \ldots, v_5)$, $L = (v_6, \ldots, v_{10})$, and $\beta_1 = 2\beta_2$. Let $\alpha = 4\beta_1 + 5\beta_2 = 13\beta_2$ (this assumption corresponds to the *minimum bandwidth* constraint, and is known as the MBR point of the storage-bandwidth tradeoff curve of [1]). By Lemma 3, the value of the minimum cut is $52\beta_2$, and thus $M = 52\beta_2$. The task of node repair is accomplished by contacting $d = 9$ nodes, and in the worst case uses the bandwidth $13\beta_2 = 13\frac{M}{52}$.

Now we will show that under the dynamic model, it is possible to increase the file size from $52\beta_2$. Suppose that at time t a node $v_i \in U$ has failed. Assume that all the nodes in L transmit $\frac{8}{7}\beta_2$ information to CU_t and the nodes in $U \setminus \{v_i\}$ transmit $\beta_1 - \frac{\beta_2}{7}$ amount of information to CU_t. Note that the total amount of information received by CU_t is $\alpha + \frac{1}{7}\beta_2$. If at time t a node $v_i \in L$ fails, all the nodes in U transmit $\beta_1 + \frac{1}{7}\beta_2$ symbols to CU_t and the nodes in $L \setminus \{v_i\}$ transmit $\frac{6}{7}\beta_2$ symbols to CU_t. Note that the total amount of information received by CU_t is again greater than α. A straightforward calculation of the minimum cut yields that

$$\min_{\pi\in\mathscr{S}_{10}}\{C(\pi)\} = 52\beta_2 + 8\cdot\frac{1}{7}\beta_2 \geqslant 53\beta_2$$

and it is obtained when $\pi = id$. If the nodes fail with equal probability, it is clear that on average, the nodes from U, L will use a bandwidth of β_1, β_2, respectively.

The above simple procedure is in fact not optimal in terms of the file size M. Namely, according to Theorem 2, we can attain the value $M \geqslant \beta_2(52 + 31/18)$, although the analysis becomes more complicated.

In the general case we obtain the following theorem which forms the main result of this paper.

Theorem 2 *Let $\mathcal{N}$ be a storage network with $\boldsymbol{S}_i, i \geqslant 1$ chosen uniformly and independently. If the node size satisfies $\alpha \geqslant (n_1 - 1)\beta_1 + n_2\beta_2$ then*

$$\overline{\mathsf{cap}}(\mathcal{N}) \geqslant C + \tfrac{\beta_1 - \beta_2}{2} \tfrac{an_1}{n} \big(a + 1 + \tfrac{n_1 - 1}{n-1}(a-1)\big).$$

The proof uses three lemmas which are stated next.

Let $\pi \in \mathscr{S}_n$ and let $S \subset V, |S| = k$. Define a function $f_\pi : S \to \mathbb{N}$ as follows. For a node $v \in S$, $f_\pi(v)$ is the number of nodes in $S \cap L$ that appear before v in π, i.e.,

$$f_\pi(v) \triangleq \sum_{i=1}^{n} \mathbb{1}_{S\cap L}(v_i) \cdot \mathbb{1}_{[1,\pi^{-1}(v)]}(\pi^{-1}(v_i)).$$

Let $T_j, j = 1, \ldots, n-1$ be the *adjacent transposition* on π, which permutes $\pi(j)$ and $\pi(j+1)$.

Lemma 5 *Let π_t be a permutation obtained at time t and let S be a set of k active nodes selected by the DC_t. Then*

$$C_t(\pi_t) = C + \sum_{v \in S\cap U} f_\pi(v)(\beta_1 - \beta_2),$$

where C is given in Lemma 3.

Proof: First, recall that the identity permutation attains the minimum cut, i.e., if $\pi_t = \mathrm{id}$ then $C_t = C$. Let $\pi \in \mathscr{S}_n$ and let $C(\pi)$ be the corresponding capacity. We claim that $C(T_j(\pi)) \in \{C(\pi) + \beta_1 - \beta_2, C(\pi), C(\pi) - \beta_1 + \beta_2\}$.

Now we note that if $\pi = (i_1, \ldots, i_n) \in \mathscr{S}_n$ is a permutation and S is a selection of k nodes with cut $C(\pi)$ such that $v_{i_j} \in U$ and that $v_{i_{j+1}} \in L$ and both are in S, then after applying $T_{i_j}(\pi)$, there is only one new node in the sum $\sum_{v\in S\cap U} f_\pi(v)(\beta_1 - \beta_2)$, namely $v_{i_{j+1}}$. ∎

In the next lemma we bound the capacity C_t below by fixing the last a places in the permutation π_t.

Lemma 6 *For a fixed t and $0 \leqslant \ell \leqslant \min(n_1, a)$, let P_t^ℓ be the probability that π_t contains ℓ nodes from U in the last $a = k - n_1$ positions. As $t \to \infty$, we have*

$$P_t^\ell \to \binom{n_1}{\ell}\binom{n_2}{a-\ell}\binom{n}{a}^{-1}.$$

Proof outline: Lemma 2 states that convergence of π_t to the uniform distribution is exponentially fast (after a certain time, the TV distance decreases by a factor of $1/e$ every n time units). Assuming the uniform distribution, we obtain the claim of the lemma. ∎

For the next lemma we need the following notation. Let $\mathscr{S}_n^\ell$ be the set of all permutations over $[n]$ with exactly ℓ numbers from U in the last a positions, i.e.,

$$\mathscr{S}_n^\ell \triangleq \{\pi \in \mathscr{S}_n : |\{\pi(n-a+1), \ldots, \pi(n)\} \cap U| = \ell\}.$$

Given $\pi = (i_1, \ldots, i_{n-a}, i_{n-a+1}, \ldots, i_n) \in \mathscr{S}_n$, let $\pi^c := (i_1, \ldots, i_{n-a}, i_n, \ldots, i_{n-a+1})$ be the *symmetric* permutation.

Lemma 7 *Let π_t be the permutation at time t and assume that π_t is distributed uniformly over $\mathscr{S}_n^\ell$. Let C be the minimum cut as in Lemma 3, then*

$$\mathbb{E}[C_t] \geqslant C + \frac{1}{2}\ell(a+\ell)(\beta_1 - \beta_2).$$

Proof: First note that the probability in Lemma 6 is determined by ℓ, n_1, n_2 and depends only on the number of the nodes from U in the last a places of π_t. We have $\sum_{\pi\in\mathscr{S}_n^\ell} \Pr(\pi|\mathscr{S}_n^\ell) C(\pi) = \frac{1}{|\mathscr{S}_n^\ell|}\sum_{\pi\in\mathscr{S}_n^\ell} C(\pi)$. To bound this sum below we fix the last a entries of the permutation. Then by Lemma 4, $C(\pi)$ is the smallest if $n_1 - \ell$ entries from U appear in the first $n_1 - \ell$ positions, followed by $n_2 - a + \ell$ entries from L (in any order). Fix the first $n - a$ entries. Again according to Lemma 4, the minimum cut will be obtained when all the ℓ nodes from U are in positions $n-a+1, n-a+2, \ldots, n-a+\ell$, and according to Lemma 5 it is equal to $C_{\min} := C + \ell^2(\beta_1 - \beta_2)$. Also, the maximum cut will be obtained when all the ℓ nodes from U are located in the last positions. This yields $C_{\max} := C + \ell a(\beta_1 - \beta_2)$.

Let $\pi \in \mathscr{S}_n^\ell$ be any permutation with $\pi(i) \in U$ for $i \in [n_1 - \ell]$. We claim that

$$C(\pi) + C(\pi^c) = 2C + \ell(a+\ell)(\beta_1 - \beta_2) = C_{\min} + C_{\max}.$$

Indeed, assume $\pi_t = \pi$ and let S be a selection of k active nodes that minimizes the cut. By Lemma 4 if there is at least one node from U in the last a places, the minimum cut will be obtained by selecting the last a places as a part of S. Moreover, if $v_i \in U$ with $\pi^{-1}(i) = n - a + m$ for some $m \in [a]$, and $f_\pi(v_i) = b$ then $|\{\pi(1), \ldots, \pi(n-a+m)\} \cap (S \cap L)| = b$. Together with the fact that $|S \cap L| = a$, this implies that $|\{\pi(n-a+1), \ldots, \pi(n-a+m)\} \cap (S \cap L)| = b - \ell$. For π^c, we obtain that $(\pi^c)^{-1}(i) = n - m + 1$ and $|\{\pi^c(n-m+1), \ldots, \pi^c(n)\} \cap L| = b - \ell$ which means that $|\{\pi^c(1), \ldots, \pi^c(n-m+1)\} \cap (S \cap L)| = a - (b - \ell)$.

By Lemma 5 we have

$$\begin{aligned} C(\pi) &= C + \sum_{v\in S\cap U} f_\pi(v)(\beta_1 - \beta_2) \\ &\geqslant C + \sum_{\substack{v\in S\cap U \\ \pi^{-1}(v)\in\{n-a+1,\ldots,n\}}} f_\pi(v)(\beta_1 - \beta_2). \end{aligned}$$

For π^c we obtain

$$\begin{aligned} C(\pi^c) &\geqslant C + \sum_{\substack{v\in S\cap U \\ (\pi^c)^{-1}(v)\in\{n-a+1,\ldots,n\}}} f_{\pi^c}(v)(\beta_1 - \beta_2) \\ &= C + \sum_{\substack{v\in S\cap U \\ (\pi^c)^{-1}(v)\in\{n-a+1,\ldots,n\}}} (a - (f_\pi(v) - \ell))(\beta_1 - \beta_2). \end{aligned}$$

This implies that $C(\pi)+C(\pi^c)\geqslant 2C+\ell(a+\ell)(\beta_1-\beta_2)$.

Note that for every $\pi\in\mathscr{S}_n^\ell$, the symmetric permutation $\pi^c\in\mathscr{S}_n^\ell$ and that $(\pi^c)^c=\pi$. Thus, we have

$$\sum_{\pi\in\mathscr{S}_n^\ell}\Pr(\pi)C(\pi)=\frac{1}{|\mathscr{S}_n^\ell|}\frac{1}{2}\sum_{\pi\in\mathscr{S}_n^\ell}C(\pi)+C(\pi^c)\geqslant C+\frac{1}{2}\ell(a+\ell)(\beta_1-\beta_2),$$

which concludes the proof. ∎

We can now complete the proof of Theorem 2.

Proof of Theorem 2: By Lemma 4 we can assume $k>n_1$. Consider $\mathbb{E}[C_t]$ for t large enough. Since $(\mathscr{S}_n^\ell)_\ell$ partition the set $\mathscr{S}_n$ we have

$$\begin{aligned}\mathbb{E}[C_t]&=\sum_{\pi_t\in\mathscr{S}_n}\Pr(\pi_t)C_t(\pi_t)\\&=\sum_{\ell=0}^{\min\{a,n_1\}}\sum_{\pi_t\in\mathscr{S}_n^\ell}\Pr(\pi_t|\mathscr{S}_n^\ell)\Pr(\mathscr{S}_n^\ell)C_t(\pi_t)\\&=\sum_{\ell=0}^{\min\{a,n_1\}}\Pr(\mathscr{S}_n^\ell)\sum_{\pi_t\in\mathscr{S}_n^\ell}\Pr(\pi_t|\mathscr{S}_n^\ell)C_t(\pi_t)\\&\overset{(a)}{\geqslant}\sum_{\ell=0}^{\min\{a,n_1\}}\Pr(\mathscr{S}_n^\ell)\left(C+\frac{1}{2}\ell(a+\ell)(\beta_1-\beta_2)\right)\\&\overset{(b)}{=}C+\sum_{\ell=0}^{n_1}\binom{n_1}{\ell}\binom{n_2}{a-\ell}\binom{n}{a}^{-1}\frac{\ell(a+\ell)(\beta_1-\beta_2)}{2},\end{aligned}$$

where (a) follows from Lemma 7 and (b) follows from Lemma 6. The final expression is obtained by repeated use of the Vandermonde convolution formula. ∎

Appendix: Proof of Lemma 4

Proof: If $k\leqslant n_1$ the result follows immediately from [3]. Indeed, for any permutation π_t, choosing k nodes from U will yield the minimum possible cut.

In the case $k>n_1$, we formulate our question as a dynamic programming problem. Assume that π_t is a fixed permutation that represents the order of the failed nodes. We will consider the information flow graph $\mathcal{X}_t$ and show that the cut is minimized when all the nodes from U are selected.

Consider a k-step procedure which in each step selects one node from $\mathcal{A}_t$. Each step entails a cost. If a node $v_{i_t}^t$ was chosen, the cost is defined as the added capacity values of the in-edges of CU_t that were not cut-off bu previously selected nodes. Our goal is to choose k nodes that minimize the total cost and hence minimize the cut between $\bigcup_{j=-1}^t\mathcal{A}_j\setminus\mathcal{A}_t$ and DC_t.

Now consider the sub-problem at step $k-1$, where the DC_t has already chosen $k-1$ nodes $(v_{i_1}^{t_1},\ldots,v_{i_{k-1}}^{t_{k-1}})$ and we are to choose the last node. Assume that the chosen nodes are ordered according to their appearance in the permutation, i.e., $t_1\leqslant t_2\leqslant\ldots\leqslant t_{k-1}$. Let $v_{j_1}^{t'_1},\ldots,v_{j_m}^{t'_m}\in U$ be nodes that were not selected up to step $k-1$, i.e.,

$$\{v_{j_1}^{t'_1},\ldots,v_{j_m}^{t'_m}\}\cap\{v_{i_1}^{t_1},\ldots,v_{i_{k-1}}^{t_{k-1}}\}=\emptyset.$$

Assume also that $t'_1\leqslant t'_2\leqslant\ldots\leqslant t'_m$. We show that choosing $v_{j_1}^{t'_1}$ accounts for the minimum cut. First, we show that choosing $v_{j_1}^{t'_1}$ minimizes the cut over all other nodes from U. Denote by C_{k-1} the total cost (or the cut) in step $k-1$. Fix $2\leqslant\ell\in[m]$ and note that since $t'_1\leqslant t'_\ell$ we may write $t_1\leqslant\ldots\leqslant t_{r_1}\leqslant t'_1\leqslant t_{r_1+1}\leqslant\ldots\leqslant t_{r_\ell}\leqslant t'_\ell\leqslant t_{\ell+1}\leqslant\ldots\leqslant t_k$. Let $C(j_1),C(j_\ell)$ be the cut values if we choose $v_{j_1}^{t'_1},v_{j_\ell}^{t'_\ell}$, respectively. Choosing $v_{j_1}^{t'_1}$ by the DC_t changes C_{k-1} as follows. First, due to its location, it will add $n_1\beta_1+n_2\beta_2-\sum_{d=1}^{r_1}\beta_{i_d}$. Next, it will reduce by β_1 the contribution of each of the nodes $v_{i_{r_1+1}}^{t_{r_1+1}},\ldots,v_{i_k}^{t_k}$ to the cost. Therefore, we obtain $C(j_1)=C_{k-1}+(n_1-1)\beta_1+n_2\beta_2-\sum_{d=1}^{r_1}\beta_{i_d}-(k-1-r_1)\beta_1$. Following the same steps for $C(j_\ell),\ell\geqslant 2$ we obtain $C(j_\ell)=C_{k-1}+(n_1-1)\beta_1+n_2\beta_2-\sum_{d=1}^{r_\ell}\beta_{i_d}-(k-1-r_\ell)\beta_1$. Hence, we have $C(j_1)-C(j_\ell)=\sum_{d=1}^{r_\ell-r_1}\beta_{i_{r_1+d}}-(r_\ell-r_1)\beta_1$ which is non-positive. Now we show that $v_{j_1}^{t'_1}$ minimizes the cut over a selection of any node v_ℓ from L. We divide the argument into 2 cases:

1) Assume that $\pi_t^{-1}(\ell)\leqslant\pi_t^{-1}(j_1)$. Again, let $C(\ell),C(j_1)$ be the cut values if we choose $v_\ell,v_{j_1}^{t'_1}$, respectively. Then $C(j_1)-C(\ell)=(k-\pi_t(j_1))(\beta_2-\beta_1)+(\pi_t(j_1)-\pi_t(\ell))\beta_2-\sum_{d=1}^{\pi_t(j_1)-\pi_t(\ell)}\beta_{i_{\ell+d}}$, which is non-positive.
2) Assume $\pi_t^{-1}(\ell)\geqslant\pi_t^{-1}(j_1)$. This case follows immediately by noticing that choosing $v_{j_1}^{t'_1}$ minimizes the cut even if $v_\ell\in U$.

Following the principle of optimality [8, Ch. 1.3], it is clear that we first need to choose all the nodes from U and then choose nodes from L. ∎

References

[1] A. Dimakis, P. B. Godfrey, Y. Wu, M. J. Wainwright, and K. Ramchandran, "Network coding for distributed storage systems," *IEEE Trans. Inf. Theory*, vol. 56, no. 9, pp. 4539–4551, 2010.

[2] V. R. Cadambe, S. A. Jafar, H. Maleki, K. Ramchandran, and C. Suh, "Asymptotic interference alignment for optimal repair of mds codes in distributed storage." *IEEE Trans. Inf. Theory*, vol. 59, no. 5, pp. 2974–2987, 2013.

[3] S. Akhlaghi, A. Kiani, and M. R. Ghanavati, "Cost-bandwidth tradeoff in distributed storage systems," *Computer Communications*, vol. 33, no. 17, pp. 2105–2115, 2010.

[4] J. Y. Sohn, B. Choi, S. W. Yoon, and J. Moon, "Capacity of clustered distributed storage," *IEEE Trans. Inf. Theory*, vol. 65, no. 1, pp. 81–107, 2019.

[5] A. M. Kermarrec, N. L. Scouarnec, and G. Straub, "Repairing multiple failures with coordinated and adaptive regenerating codes," in *Int. Symp. on Network Coding (NetCod)*. IEEE, 2011, pp. 1–6.

[6] G. Grimmett and D. Stirzaker, *Probability and Random Processes*, 3rd ed. Oxford Univ. Press, 2001.

[7] D. Aldous and P. Diaconis, "Shuffling cards and stopping times," *The American Mathematical Monthly*, vol. 93, no. 5, pp. 333–348, 1986.

[8] D. P. Bertsekas, *Dynamic programming and optimal control*. Athena scientific Belmont, MA, 2005, vol. 1.