Service Rate Region of Content Access from Erasure Coded Storage

Sarah E. Anderson*, Ann Johnston†, Gauri Joshi¶, Gretchen L. Matthews‡, Carolyn Mayer§, and Emina Soljanin‖
*University of St. Thomas, St. Paul, Minnesota, USA, ande1298@stthomas.edu
†Penn State University, University Park, Pennsylvania, USA, abj5162@psu.edu
¶Carnegie Mellon University, Pittsburgh, PA, USA, gaurij@andrew.cmu.edu
‡Clemson University, Clemson, South Carolina, USA, gmatthe@clemson.edu
§Worcester Polytechnic Institute, Worcester, Massachusetts, USA, cdmayer@wpi.edu
‖Rutgers University, New Brunswick, NJ, USA, emina.soljanin@rutgers.edu

***Abstract*—We consider storage systems in which K files are stored over N nodes. A node may be systematic for a particular file in the sense that access to it gives access to the file. Alternatively, a node may be coded, meaning that it gives access to a particular file only when combined with other nodes (which may be coded or systematic). Requests for file f_k arrive at rate λ_k, and we are interested in the rate that can be served by a particular system. In this paper, we determine the set of request arrival rates for the a 3-file coded storage system. We also provide an algorithm to maximize the rate of requests served for file K given $\lambda_1, \ldots, \lambda_{K-1}$ in a general K-file case.**

I. Introduction

The explosive growth in the amount of data stored in the cloud calls for new techniques to make cloud infrastructure fast, reliable, and efficient. Moreover, applications that access this data from the cloud are becoming increasingly interactive. Thus, in addition to providing reliability against node failures, service providers must be able to serve a large number of users simultaneously.

Content files are typically replicated at multiple nodes to cope with node failures. These replicas can also be used to serve a larger volume of users. To adapt to changes in popularity of content files, service providers can increase or decrease the number of replicates for each file, a strategy that has been widely used in content delivery networks [1]. The use of erasure coding, instead of replication, to improve the availability of content is not yet fully understood. Using erasure codes has been shown to be effective in reducing the delay in accessing a file stored on multiple servers [2]–[4]. However, only a few works have studied their use to store multiple files. Some recent works [5] have proposed new classes of erasure codes to store multiple files that allow a file to be read from from disjoint sets of nodes. Other works [6], [7] study the delay reduction achieved using these codes.

Besides download latency, it has recently been recognized that another important metric for the availability of stored data is the service rate [8]–[10]. Maximizing the service rate (or the throughput) of a distributed system helps support a large number of simultaneous system users. Rate-optimal strategies are also latency-optimal in high traffic. Thus, maximizing the service rate also reduces the latency experienced by users, particularly in highly contending scenarios.

This paper is one of the first to analyze the *service rate region* of a coded storage system. We consider distributed storage systems in which data for K files is to be stored across N nodes. A request for one of the files can be either sent to a systematic node or to one of the repair groups. We seek to maximize such systems' service rate region, that is, the set of request arrival rates for the K files that can be supported by a coded storage system.

The problem addressed in this paper should not be confused with the related problem of caching and pre-fetching of popular content at edge devices [11]. Caching benefits for the network are measured in reduction in the backhaul traffic it enables. Quality of service to the user measures include cache hit ratio and cache hit distance. Rather than with the backhaul, this paper is concerned with the access part of the network, namely, with potential service rate increase through work provided, jointly and possibly redundantly, by multiple network edge devices. Consequently, instead of measuring e.g., content download performance by the likelihood of an individual cache hit or cache memory and bandwidth usage, we strive to ensure that multiple caches are jointly in possession of content and can deliver it fast to multiple simultaneous users.

In [9], the achievable service rate region was found for some common classes of codes, such as maximum-distance-separable (MDS) codes and simplex codes. That paper also determined the service rate region when $K = 2$, with arbitrary numbers of systematic and coded nodes. We generalize this service rate region result from $K = 2$ files to $K = 3$ files and provide an algorithm to maximize the requests served for a given file with general K. The paper begins with preliminary notions given in Sec. II. Sec. III addresses the general K case where all nodes are coded, and Sec. IV addresses the $K = 3$ case. We return to the general case in Sec. V.

II. Preliminaries

Suppose files $f_1, \ldots, f_K$ are stored across a system that consists of N nodes labeled $1, \ldots, N$. For $k \in$

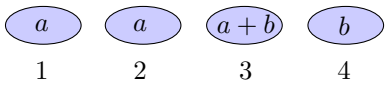


Fig. 1. A possible way to store two files on $N = 4$ nodes.

$[K] := \{1, \ldots, K\}$, there is a collection of minimal sets $R_{k1}, \ldots, R_{k\gamma_k} \subseteq [N]$ that each correspond to a set of nodes that gives access to file f_k. Each such minimal set of nodes is called an f_k–repair group.

Example 1. *Fig. 1 shows one possible way to store two files, a and b, across four nodes. In this system, the a-repair groups are $\{1\}, \{2\}$, and $\{3, 4\}$. The b-repair groups are $\{4\}, \{1, 3\}$, and $\{2, 3\}$.*

For $(i, j) \in [\gamma_k] \times [N]$, define the function

$$\delta_k(i, j) := \begin{cases} 1, & \text{if node } j \text{ is in the } f_k\text{–repair group } R_{ki}, \\ 0, & \text{else.} \end{cases} \quad (1)$$

Suppose that when a request for file f_k is received, that request is sent at random to an f_k–repair group according to a splitting strategy with $\alpha_{ki} \geq 0$ denoting the fraction of requests sent to repair group R_{ki}, so that for each $k \in [K]$,

$$\sum_{i \in [\gamma_k]} \alpha_{ki} = 1. \quad (2)$$

Let the demand for file f_k be λ_k, so the arrival of requests for file f_k to the storage system queue is Poisson with rate λ_k, and let $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_K)$ record the demand for files $f_1, \ldots, f_K$.

The average rate that file requests arrive at a storage system node depends both on the splitting strategy for file requests and on the demand $\boldsymbol{\lambda}$. More precisely, the average rate that file requests are received at node $j \in [N]$ is

$$\sum_{k \in [K]} \sum_{i \in [\gamma_k]} \alpha_{ik} \delta_k(i, j) \, \lambda_k. \quad (3)$$

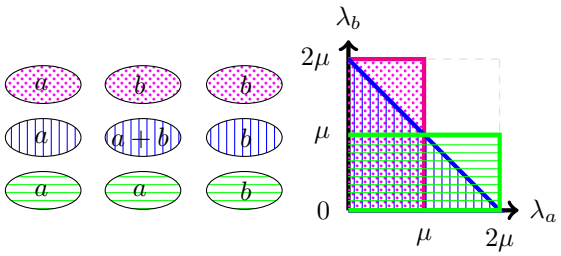
Let μ_j denote the average rate of resolving received file requests at node j. Whenever demand is such that at least one node j of the storage system receives requests at an average rate in excess of its μ_j, the storage system queue will have a tendency to grow. With this in mind, it is appropriate to call μ_j the service rate of node j. We will consider uniform systems for which $\mu_j = 1$ for $j = 1, \ldots, N$. If, at demand $\boldsymbol{\lambda}$, there exists a splitting strategy under which no storage system node receives requests at a rate in excess of its service rate, then $\boldsymbol{\lambda}$ is said to be in the achievable service rate region of the storage system. More formally, the storage system's achievable service rate region $\mathcal{S}$ is the set of all $\boldsymbol{\lambda} \in \mathbb{R}^K_{\geq 0}$ such that there exists a splitting strategy with

$$\sum_{k \in [K]} \sum_{i \in [\gamma_k]} \alpha_{ki} \delta_k(i, j) \, \lambda_k \leq \mu_j, \quad \text{for all } j \in [N]. \quad (4)$$

For any $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_K) \in \mathbb{R}^K_{\geq 0}$, denote by $\boldsymbol{\lambda}_{\widehat{k}}$ the $(K-1)$-tuple $(\lambda_1, \ldots, \lambda_{k-1}, \lambda_{k+1}, \ldots \lambda_K)$, and for $x \in \mathbb{R}_{\geq 0}$ let $\boldsymbol{\lambda}_{\widehat{k}} \times \{x\} := (\lambda_1, \ldots, \lambda_{k-1}, x, \lambda_{k+1}, \ldots \lambda_K)$. If $\boldsymbol{\lambda} \in \mathcal{S}$, then the same splitting strategy whose existence is guaranteed by (4) is also sufficient to give $\boldsymbol{\lambda}' \in \mathcal{S}$ for every $\boldsymbol{\lambda}'$ satisfying for all $k \in [K]$, $\lambda'_k \leq \lambda_k$. Thus, given any pair $\boldsymbol{\lambda}_{\widehat{k}} \times \{0\} \in \mathcal{S}$ and $\boldsymbol{\lambda}_{\widehat{k}} \times \{\lambda_k\} \in \mathcal{S}$, the entire interval $\boldsymbol{\lambda}_{\widehat{k}} \times [0, \lambda_k]$ is in $\mathcal{S}$. Moreover, for any storage system (regardless of its coding), if $\boldsymbol{\lambda}$ is such that the demand for any file f_k is in excess of $N \cdot \max_{j \in [N]}\{\mu_j\}$, then under all possible assignment strategies (4) is violated for at least one node j, and so $\boldsymbol{\lambda}_{\widehat{k}} \times \{x\}$ is not in $\mathcal{S}$ for any $x > N \cdot \max_{j \in [N]}\{\mu_j\}$ and $\boldsymbol{\lambda}_{\widehat{k}} \in \mathbb{R}^{K-1}_{\geq 0}$. In this way, $\mathcal{S}$ is a non-empty, closed, and bounded subset of $\mathbb{R}^K_{\geq 0}$. Therefore, given any $\boldsymbol{\lambda}_{\widehat{k}} \times \{0\} \in \mathcal{S}$, there exists a maximal value of λ_k such that $\boldsymbol{\lambda}_{\widehat{k}} \times [0, \lambda_k] \subset \mathcal{S}$ and $\boldsymbol{\lambda}_{\widehat{k}} \times \{\lambda'_k\} \notin \mathcal{S}$ for any $\lambda'_k > \lambda_k$. When $k = K$, we call this maximal value $L(\boldsymbol{\lambda}_{\widehat{K}})$. In this notation, the service rate region of any storage system can be described as:

$$\mathcal{S} = \{\boldsymbol{\lambda}_{\widehat{K}} \times [0, L(\boldsymbol{\lambda}_{\widehat{K}})] : (\lambda_1, \ldots, \lambda_{K-1}, 0) \in \mathcal{S}\}. \quad (5)$$

Example 2. *Three examples of how two files, a and b may be stored across three nodes are shown on the below on the left. The resulting service service rate regions for each system are shown below on the right.*



Coding schemes that use a mixture of replication and MDS coding are not conventional. However, if the service rate region is used as a performance metric, then a combination of coded and systematic nodes has been shown to be beneficial [6], [9]. In this paper, we consider storage systems for K files whose coded nodes satisfy the following three conditions:

1) Each K–subset of coded nodes forms an f_k–repair group for every $k \in [K]$.
2) No subset of $k < K$ coded nodes forms an f_k–repair group, for any $k \in [K]$.
3) With addition of systematic nodes for any n distinct files (naturally, $n < K$) every $(K-n)$–subset of coded nodes from the core completes these systematic nodes to form an f_k–repair group for every $k \in [K]$.

We say that such a system has an MDS core. We consider situations with uniform node capacities $\mu = \mu_1 = \cdots = \mu_N$.

For convenience, we use C to denote the number of coded nodes in such a core. When systematic nodes are also present, we use N_k to denote the number of systematic nodes for file f_k. In this way, the total number of nodes in a storage system for K files that has an MDS core is $N = C + \sum_{k=1}^K N_k$.

III. All coded nodes

We begin by considering an MDS K-file core where there are no systematic nodes in the system. In this situation, all nodes form a repair group for each file, and K nodes are required to recover any file.

Theorem 1. *Assume $N_1 = \cdots = N_K = 0$. If there are $C > K-1$ coded nodes, then the achievable service rate region $\mathcal{S}$ is the set of all $\boldsymbol{\lambda}$ with $\sum_{i=1}^{K} \lambda_i \leq \frac{C}{K}\mu$, and so $L(\lambda_1, \ldots, \lambda_{K-1}) = \frac{C}{K}\mu - \sum_{i=1}^{K-1} \lambda_i$. If there are $C \leq K-1$ coded nodes, then $\mathcal{S}$ is the point $(0, \ldots, 0)$.*

Proof. If $C \leq K-1$, then no file can be recovered and the service rate region is the point $(0, \ldots, 0)$.

Assume $C > K-1$. Note that since every repair group requires K nodes, the total demand that can be served is bounded above by $\frac{C\mu}{K}$. For each file, there are a total of $\binom{C}{K}$ repair groups, and each node is in $\binom{C-1}{K-1}$ repair groups. By sending demand $\frac{\mu}{\binom{C-1}{K-1}}$ to each repair group, requests to each node occur at the service rate and the system can serve demand $\frac{\binom{C}{K}}{\binom{C-1}{K-1}}\mu = \frac{C}{K}\mu$. Since this demand can be for any file, the service rate region is $\sum_{i=1}^{K} \lambda_i \leq \frac{C}{K}\mu$. Therefore, the maximum achievable λ_K is

$$\lambda_K = L(\lambda_1, \ldots, \lambda_{K-1}) = \frac{C}{K}\mu - \sum_{i=1}^{K-1} \lambda_i.$$

□

The two file case is considered in [9]. The situation becomes increasingly complex depending on the number of files K in the system. In the next section, we consider $K = 3$.

IV. Three files

In this section, we consider the service rate region of storage systems for 3 files with MDS cores. As a corollary to Theorem 1, we obtain the service rate region for the case when there are no systematic nodes, which is represented in Fig. 2. Note that when the demand for one file is zero, then this may be considered a system with only two files. For example, if $\lambda_3 = 0$, then the maximum achievable λ_2 is $\lambda_2 = \frac{C}{3}\mu - \lambda_1$, which is the region shaded in Fig. 2.

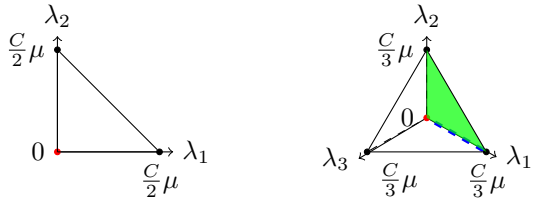


Fig. 2. Achievable service rate regions of all-coded-node systems with 2 files (left) or 3 files (right).

We now consider storage systems that have both coded nodes and systematic nodes. Suppose that a coded storage system has C coded nodes and N_i systematic file f_i nodes, $i = 1, 2, 3$. Note that a systematic repair node may be in a repair group with a single node (serving requests for the file it stores) or three nodes (serving requests for any other file). Any repair group using a coded node contains three nodes. For $i = 1, 2$, if $r_i \leq N_i\mu$ requests for file f_i are served using systematic f_i nodes (and any other demand for file f_i is served using a repair group of three nodes), then the total demand that can be served is bounded above by

$$D := r_1 + r_2 + \frac{(N_1\mu - r_1) + (N_2\mu - r_2) + C\mu}{3} + N_3\mu.$$

Given demand λ_1 for file f_1 and λ_2 for file f_2, the rate of requests that may be served for file f_3 is bounded above by $\max\{D - \lambda_1 - \lambda_2, 0\}$. This is maximized when $r_i = \min\{\lambda_i, N_i\mu\}$ for $i = 1, 2$. The splitting strategy in the proof of the following theorem meets this bound.

Theorem 2. *Assume there are N_1, N_2, and N_3 systematic nodes for files f_1, f_2, and f_3, respectively, and C coded nodes. Assume $\lambda_1 + \lambda_2 \leq \mu N_1 + \mu N_2 + \frac{C}{3}\mu$ and $C \geq \max\left(3, N_1 - \frac{\lambda_1}{\mu}, N_2 - \frac{\lambda_2}{\mu}\right)$. Then $\mathcal{S}$ has $L(\lambda_1, \lambda_2) =$*

$$\begin{cases} (\frac{C}{3} + \frac{N_1}{3} + \frac{N_2}{3} + N_3)\mu - \frac{\lambda_1}{3} - \frac{\lambda_2}{3}, & 0 \leq \frac{\lambda_i}{\mu} \leq N_i,\ i = 1, 2 \\ (\frac{C}{3} + N_1 + \frac{N_2}{3} + N_3)\mu - \lambda_1 - \frac{\lambda_2}{3}, & N_1 < \frac{\lambda_1}{\mu} \leq N_1 + N_2 + \frac{C}{3}, \\ & 0 \leq \frac{\lambda_2}{\mu} \leq N_2 \\ (\frac{C}{3} + \frac{N_1}{3} + N_2 + N_3)\mu - \frac{\lambda_1}{3} - \lambda_2, & 0 \leq \frac{\lambda_1}{\mu} \leq N_1, \\ & N_2 < \frac{\lambda_2}{\mu} \leq N_1 + N_2 + \frac{C}{3} \\ (\frac{C}{3} + N_1 + N_2 + N_3)\mu - \lambda_1 - \lambda_2, & N_1 < \frac{\lambda_1}{\mu} \leq N_1 + N_2 + \frac{C}{3}, \\ & N_2 < \frac{\lambda_2}{\mu} \leq N_1 + N_2 + \frac{C}{3} \end{cases}$$

Proof. Consider a system with N_1, N_2, and N_3 systematic nodes for files f_1, f_2, and f_3 and C coded nodes.

Step 1: Send requests to systematic nodes at the service rate to serve demand for files f_1 and f_2, as possible. If any f_i $(i = 1, 2)$ systematic nodes remain available, distribute remaining file f_i demand uniformly across those nodes.

Example 3. *Consider a 3-file system with $N_1 = 3$, $N_2 = 1$, $N_3 = 1$, and $C = 3$.*

If $\lambda_1 = \frac{3}{2}\mu$ and $\lambda_2 = 2\mu$ then μ requests for f_1 will be served by one of the f_1 systematic nodes, and the remaining $\frac{1}{2}\mu$ requests for f_1 will be split between the other 2 systematic nodes. Also, μ requests for f_2 will be served by the f_2 systematic node. After Step 1, the remaining demand for f_1 is 0 and the remaining demand for file f_2 is μ. In the system, there are now two systematic f_1 nodes that can handle an additional $\frac{3}{4}\mu$ requests as well as one systematic f_3 node and three coded nodes each with available service rate μ.

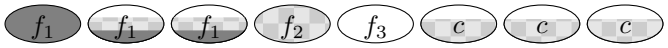
At the end of Step 1, if $\lambda_i \leq \mu N_i$ for $i = 1$ or 2, then there will be $N_i' = N_i - \lfloor \frac{\lambda_i}{\mu} \rfloor$ systematic nodes remaining available for f_i, each with service rate reduced to $\mu_i' = \mu - \frac{\lambda_i - \lfloor \frac{\lambda_i}{\mu} \rfloor \cdot \mu}{N_i'}$. Since $\lambda_i \leq \mu N_i$, the remaining demand for file f_i is $\lambda_i' = 0$.

If $\lambda_i \geq \mu N_i$ for $i = 1$ or 2, we exhaust every f_i systematic node. The remaining demand for file f_i is then $\lambda_i' = \lambda_i - \mu N_i$, and $N_i' = 0$ systematic f_i nodes remain.

Step 2: Serve any remaining demand for files f_1 and f_2. Finally, serve demand for file f_3.

Example 4. *Consider the system in Example 3. In Step 2 we want to serve the remaining requests for file f_2 in a way that maximizes the requests that can be handled for f_3. In particular, we will reserve the use of systematic f_3 nodes for*

accessing file f_3. Note that there are $2\cdot\binom{3}{2}=6$ repair groups for file f_2 that involve one systematic f_1 node and two coded nodes. If we send $\frac{\mu}{6}$ requests for file f_2 to each of these repair groups, then all the requests for file f_2 are served, each f_1 systematic node can serve $\frac{\mu}{4}$ more requests (as each f_1 node is in 3 repair groups) and each coded node can serve $\frac{\mu}{3}$ more requests (as each coded node is in 4 repair groups).



Finally, requests for f_3 may be served. Sending $\frac{\mu}{12}$ requests to each of the 6 repair groups with one f_1 node and two coded nodes exhausts each f_1 node and each coded node. The full service rate of the systematic f_3 node may also be used to serve requests for f_3. Thus a total of $6\cdot\frac{\mu}{12}+\mu=\frac{3}{2}\mu$ requests for f_3 may be served.

How requests are served in Step 2 depends on the demand and number of systematic nodes for files f_1 and f_2. Let λ be the total demand for files f_1 and f_2 that remains after Step 1; that is, $\lambda=\lambda_1'+\lambda_2'$.

Case 1 *($0\le\lambda_1\le\mu N_1$, $0\le\lambda_2\le\mu N_2$):* In this case, $\lambda=0$, so all available system resources may be used to serve f_3 demand. The full service rate of file f_3 systematic nodes may be used, serving demand μN_3 for file f_3. Let σ be a permutation on $\{1,2\}$ such that $\frac{\mu'_{\sigma(1)}}{N'_{\sigma(2)}}\le\frac{\mu'_{\sigma(2)}}{N'_{\sigma(1)}}$.

There are $N_1'N_2'C$ f_3–repair groups with a systematic node for each of f_1 and f_2, and one coded node. Recall, $C\ge\max\left(3,N_1-\frac{\lambda_1}{\mu},N_2-\frac{\lambda_2}{\mu}\right)$. Since $C\ge N_{\sigma(1)}-\frac{\lambda_{\sigma(1)}}{\mu}$,

$$\mu'_{\sigma(1)}N'_{\sigma(1)}=\mu\left(N_{\sigma(1)}-\frac{\lambda_{\sigma(1)}}{\mu}\right)$$

demand for f_3 can be served by sending $\frac{\mu'_{\sigma(1)}}{N'_{\sigma(2)}C}$ demand to each of these repair groups. The service rate of each $f_{\sigma(1)}$ is reduced to 0, while $f_{\sigma(2)}$ systematic nodes have $\mu''_{\sigma(2)}=\mu'_{\sigma(2)}-\frac{\mu'_{\sigma(1)}}{N'_{\sigma(2)}C}N'_{\sigma(1)}C=\mu'_{\sigma(2)}-\frac{\mu'_{\sigma(1)}}{N'_{\sigma(2)}}N'_{\sigma(1)}$, and coded nodes have $\mu'_C=\mu-\frac{\mu'_{\sigma(1)}}{N'_{\sigma(2)}C}N'_{\sigma(1)}N'_{\sigma(2)}=\mu-\frac{\mu'_{\sigma(1)}}{C}N'_{\sigma(1)}$.

There are $N'_{\sigma(2)}\binom{C}{2}$ f_3–repair groups with one of the remaining systematic file $f_{\sigma(2)}$ nodes and 2 coded nodes. Since $C\ge N_{\sigma(2)}-\frac{\lambda_{\sigma(2)}}{\mu}$, similarly to before, we can serve

$$\mu''_{\sigma(2)}N'_{\sigma(2)}=\mu\left(\left(N_{\sigma(2)}-\frac{\lambda_{\sigma(2)}}{\mu}\right)-\left(N_{\sigma(1)}-\frac{\lambda_{\sigma(1)}}{\mu}\right)\right)$$

demand for file f_3 by sending demand equally to each of these f_3–repair groups. Each coded node has remaining service rate $\mu''_C=\mu'_C-\frac{\mu''_{\sigma(2)}}{\binom{C}{2}}(C-1)N'_{\sigma(2)}$, and no systematic f_1,f_2 nodes remain available.

Since $C\ge3$, as in the case in Theorem 1 with C coded nodes and no systematic nodes, the service rate μ''_C of these coded nodes can be used to serve $\frac{C}{3}\mu''_C$ demand for file f_3.

Thus, the maximum achievable λ_3 is $L(\lambda_1,\lambda_2)$

$$\begin{aligned}&=\frac{C}{3}\mu''_C+\mu''_{\sigma(2)}N'_{\sigma(2)}+\mu'_{\sigma(1)}N'_{\sigma(1)}+\mu N_3\\&=\frac{1}{3}\left(C\mu+\mu N_{\sigma(2)}-\lambda_{\sigma(2)}+\mu N_{\sigma(1)}-\lambda_{\sigma(1)}\right)+\mu N_3.\end{aligned}$$

Similar arguments can be used for **Case 2**: $\mu N_1<\lambda_1\le\mu N_1+\mu N_2+\frac{C}{3}\mu$, $0\le\lambda_2\le\mu N_2$ and **Case 3**: $0\le\lambda_1\le\mu N_1$, $\mu N_2<\lambda_2\le\mu N_1+\mu N_2+\frac{C}{3}\mu$ (see Example 4).

Case 4 *($\mu N_1<\lambda_1\le\mu N_1+\mu N_2+\frac{C}{3}\mu$, $\mu N_2<\lambda_2\le\mu N_1+\mu N_2+\frac{C}{3}\mu$):* In this case, all available repair groups consist entirely of coded nodes. Since demand μN_i for file f_i ($i=1,2$) was satisfied in Step 1, the remaining total demand for files f_1 and f_2 is $\lambda<\frac{C}{3}\mu$. Since $C\ge3$, this can be served by sending demand equally to every coded repair group. The coded nodes' remaining ability to service can be used for file f_3. Thus, the maximum achievable λ_3 is

$$\begin{aligned}L(\lambda_1,\lambda_2)&=\frac{C}{3}\mu-\lambda+\mu N_3\\&=\frac{C}{3}\mu-(\lambda_1-\mu N_1+\lambda_2-\mu N_2)+\mu N_3.\end{aligned}$$

□

Note that $L(\lambda_1,\lambda_2)$ can be found for systems with $C<3$ coded nodes in a similar way. When $C<3$, all repair groups must contain systematic nodes for at least $3-C$ distinct files.

V. MDS K-FILE CORES

Theorem 2 may be generalized to provide an algorithm for maximizing λ_k for the general K-file case. Assume we have an MDS K-file core with $N_1,N_2,\ldots,N_K$ systematic nodes for files $f_1,f_2,\ldots,f_K$, respectively, and C coded nodes, with demand $\lambda_1,\lambda_2,\ldots,\lambda_{K-1}$ for files $f_1,f_2,\ldots,f_{K-1}$. As in Theorem 2, we again assume $\lambda_1+\ldots+\lambda_{K-1}\le\mu N_1+\ldots+\mu N_{K-1}+\frac{C}{K}\mu$. Our goal is to identify the maximal file f_K request rate that can be served.

We can first serve file $f_1,f_2,\ldots,f_{K-1}$ demand using their respective systematic nodes. This process is analogous to Step 1 in Theorem 2. Note, in this algorithm, the same demand is sent to every file f_i systematic node, and also to every coded node, so we can let μ_i and μ_C represent the updated service rate of systematic file f_i nodes and coded nodes, respectively.

We can then serve any remaining total demand $\lambda=\lambda_1+\ldots+\lambda_{K-1}$ using K-tuples of coded and systematic nodes. This is analogous to Step 2 in Theorem 2. Let $K':=\sum_{i=1}^{K-1}\operatorname{sgn}(N_i)$ denote the number of files (excluding file f_K) for which the system contains systematic nodes. There are $\left(\prod_{\{i=1\mid N_i>0\}}^{K-1}N_i\right)\binom{C}{K-K'}$ repair groups with K' systematic nodes and $K-K'$ coded nodes. Letting m be the index minimizing $N_i\mu_i$ for positive $N_i\mu_i$, we can serve demand μ_mN_m by sending $\frac{\mu_m}{\left(\prod_{\{i=1\mid N_i>0\text{ and }i\ne m\}}^{K-1}N_i\right)\binom{C}{K-K'}}$ demand to each of these repair groups. This exhausts file f_m systematic

nodes, while file f_j systematic nodes ($j \neq m$, $N_j > 0$, $1 \leq j \leq K-1$) have reduced service rate

$$\mu_j \quad - \quad \frac{\mu_m \left(\prod_{\{i=1 \mid N_i>0 \text{ and } i\neq j\}}^{K-1} N_i\right) \binom{C}{K-K'}}{\left(\prod_{\{i=1 \mid N_i>0 \text{ and } i\neq m\}}^{K-1} N_i\right) \binom{C}{K-K'}}, \quad (6)$$

which is $\mu_j - \frac{\mu_m N_m}{N_j}$. The remaining coded nodes have reduced service rate

$$\mu_C \quad - \quad \frac{\mu_m \left(\prod_{\{i=1 \mid N_i>0\}}^{K-1} N_i\right) \binom{C-1}{K-K'-1}}{\left(\prod_{\{i=1 \mid N_i>0 \text{ and } i\neq m\}}^{K-1} N_i\right) \binom{C}{K-K'}}, \quad (7)$$

which is $\mu_j - \frac{\mu_m (K-K')}{C}$. We can continue in this way until the systematic node service rate is met for all but file f_K. Then, we can use repair groups that consist entirely of coded nodes, applying Theorem 1. Once all demand for files $f_1, \ldots, f_{K-1}$ has been satisfied, we can follow a similar process to utilize any remaining system resources to serve demand for file f_K. Note, once the coded nodes have been exhausted, or if there are too few coded nodes to form a K-tuple, no demand may be satisfied using only coded nodes. We may then serve demand for file f_K using systematic file f_K nodes.

ACKNOWLEDGMENT

The initial stages of this work were performed at ICERM (Institute for Computational and Experimental Research in Mathematics) in Providence, RI. We are indebted to the organizers of the ICERM 2017 Women in Data Science and Mathematics Research Collaboration Workshop.

REFERENCES

[1] S. Borst, V. Gupta, and A. Walid, "Distributed caching algorithms for content distribution networks," in *Proc. IEEE INFOCOM*, Mar. 2010, pp. 1–9.

[2] G. Joshi, Y. Liu, and E. Soljanin, "On the delay-storage trade-off in content download from coded distributed storage systems," *IEEE JSAC*, vol. 32, no. 5, pp. 989–997, May 2014.

[3] N. Shah, K. Lee, and K. Ramachandran, "The MDS queue: Analyzing the Latency Performance of Erasure Codes," in *Proc. IEEE Int. Symp. Inform. Theory*, Jul. 2014.

[4] G. Joshi, E. Soljanin, and G. Wornell, "Efficient redundancy techniques for latency reduction in cloud systems," *ACM Trans. Modeling and Performance Evaluation of Computing Systems*, May 2017.

[5] A. Rawat, D. Papailiopoulos, A. Dimakis, and S. Vishwanath, "Locality and availability in distributed storage," in *Proc. IEEE Int. Symp. Inform. Theory*, June 2014, pp. 681–685.

[6] S. Kadhe, E. Soljanin, and A. Sprintson, "When do the availability codes make the stored data more available?" in *2015 53rd Annu. Allerton Conf. Commun., Control, and Computing*, Sept 2015, pp. 956–963.

[7] M. F. Aktas, E. Najm, and E. Soljanin, "Simplex queues for hot-data download," in *Proc. 2017 ACM SIGMETRICS/Int. Conf. Measurement and Modeling of Computer Systems*. ACM, 2017, pp. 35–36.

[8] M. Noori, E. Soljanin, and M. Ardakani, "On storage allocation for maximum service rate in distributed storage systems," in *Proc. IEEE Int. Symp. Inform. Theory (ISIT)*, July 2016, pp. 240–244.

[9] M. Aktas, S. E. Anderson, A. Johnston, G. Joshi, S. Kadhe, G. L. Matthews, C. Mayer, and E. Soljanin, "On the service capacity of accessing erasure coded content," in *Proc. Allerton Conf. Commun., Control and Computing*, Oct. 2017.

[10] G. Joshi, "Synergy via redundancy: Boosting service capacity with adaptive replication," in *Proc. IFIP Performance*, Nov. 2017.

[11] M. A. Maddah-Ali and U. Niesen, "Coding for caching: fundamental limits and practical challenges," *IEEE Commun. Mag.*, vol. 54, no. 8, pp. 23–29, 2016.

Algorithm 1 Maximize λ_K

INPUT: $\lambda_1, \lambda_2, \ldots, \lambda_{K-1}, N_1, N_2, \ldots, N_K, C, \mu$
OUTPUT: λ_K

$\lambda_K \leftarrow 0$
$\mu_C, \mu_i \leftarrow \mu$ for i from 1 to K
Step 1:
for i from 1 to $K-1$ **do**
  if $\lambda_i \leq \mu N_i$ **then**
    $\lambda_i \leftarrow 0$
    $N_i \leftarrow N_i - \left\lfloor \frac{\lambda_i}{N_i} \right\rfloor$
    $\mu_i \leftarrow \mu - \frac{\lambda_i - \left\lfloor \frac{\lambda_i}{\mu} \right\rfloor \mu}{N_i}$
  else
    $\lambda_i \leftarrow \lambda_i - \mu N_i$
    $N_i, \mu_i \leftarrow 0$
  end if
end for
Step 2:
$\lambda \leftarrow \sum_{i=1}^{K-1} \lambda_i$
$K' \leftarrow \sum_{i=1}^{K-1} \operatorname{sgn}(N_i)$
while $C > 0$ **and** $C \geq K - K'$ **do**
  if $K' > 0$ **then**
    $m \leftarrow$ the index i minimizing $N_i\mu_i, N_i\mu_i > 0$
    $l \leftarrow \min(\mu_m N_m, \mu_C C)$
    if $\lambda > 0$ **then**
      if $\lambda \geq l$ **then**
        $\lambda \leftarrow \lambda - l$
      else
        $\lambda_K \leftarrow \lambda_K + (l - \lambda)$
        $\lambda \leftarrow 0$
      end if
    else
      $\lambda_K \leftarrow \lambda_K + l$
    end if
    if $l = \lambda_m N_m$ **then**
      $\mu_C \leftarrow$ apply Equation 7
      $N_m, \mu_m \leftarrow 0$
      $K' \leftarrow K' - 1$
    else
      $\mu_C, C \leftarrow 0$
    end if
    $\mu_j \leftarrow$ apply Equation 6 if $N_j > 0$ for $1 \leq j \leq K-1$
  else
    if $\lambda > 0$ **then**
      $\mu_C \leftarrow \mu_C - \frac{\lambda}{\binom{C}{K}} \binom{C-1}{K-1}$
      $\lambda \leftarrow 0$
    end if
    $\lambda_K \leftarrow \lambda_K + \frac{C}{K} \mu_C$
    $C \leftarrow 0$
  end if
end while
$\lambda_K \leftarrow \lambda_K + \mu_K N_K$