This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSAC.2016.2545538, IEEE Journal on Selected Areas in Communications

Green Distributed Storage Using Energy Harvesting Nodes

Abdelrahman M. Ibrahim, *Student Member, IEEE,* Ahmed A. Zewail, *Student Member, IEEE,* and Aylin Yener, *Fellow, IEEE*

***Abstract*—We consider a distributed storage system where data storage nodes are equipped with energy harvesting transmitters. In particular, F files are stored over n storage nodes using regenerating codes. The main operations of the distributed storage system are serving the file requests of data collectors and repairing the content of storage nodes that fail or leave the system. Each operation has an associated energy expenditure. Under the intermittent energy arrival profile, we study the problem of maximizing the number of retrieved files given a deadline. Additionally, we consider the problem of minimizing the repair time of a failed node. Both optimization problems turn out to be equivalent to binary programs, for which we provide a tractable solution in two steps. First, we determine necessary and sufficient conditions on the harvested energy that ascertain the feasibility of retrieving (repairing) M files in T time slots. Using these conditions, we develop two algorithms that reduce the formulated optimization problems to a single feasibility problem. Then, we solve the feasibility problem using forward and backward algorithms. Additionally, we study the online setup where only causal knowledge of energy arrivals is available at the network nodes. We present numerical results on the short and long term performance of the system operations under the proposed algorithms.**

***Index Terms*—Energy harvesting, green distributed storage systems, power allocation, deadline constrained file retrieval.**

I. Introduction

Distributed Storage Systems (DSS) offer reliable and efficient storage and retrieval of data files [1]. In such systems, a data file is encoded into n pieces, each of size α, and stored over n storage nodes, i.e., each node stores one piece of the file. The file is encoded such that it can be reconstructed from the contents of any k out of the n nodes, $k < n$. This approach enhances the reliability of the file retrieval process and offers better storage efficiency than replicating the file in many locations. Furthermore, if a node fails or exits the system, a new node can join the system to take over the role of the failed one. In particular, the system initiates a repair process in which the newcomer generates a piece of size α that preserves the system functionality, by connecting to any d alive nodes, $d \leq n-1$, and downloading β bits from each of them, which results in a total repair bandwidth of $d\beta$. Since the seminal work of [1], which characterized the fundamental trade-off between the node storage capacity α and the repair bandwidth $d\beta$, extensive research effort has transpired towards designing regenerating codes, that achieve this trade-off, see for example [1]–[8], and references therein.

In the aforementioned references, node failures are the only considered dynamic in the system. In many practical scenarios, the system has other uncertainties besides the node failures, which have been considered more recently. For example, reference [9] considers a distributed storage system where the repair process is done over erasure channels. The probability of successful repair is studied and methods for maximizing it are proposed. The impact of the network topology on the repair process is considered in [10], and modeled by a transmission cost during the repair process. In particular, given the network cost matrix, a regenerating code is designed to minimize the repair cost. Reference [11] has considered a distributed storage system with multiple files and investigated the file retrieval latency under different scheduling schemes. More recent efforts also include wireless applications of distributed storage [12]–[16]. Reference [14] investigates the system performance when the repair process is done over fading channels. Reference [15] considers a system where one file is stored in a base station that serves a set of end users. Additionally, the file content is encoded using a proper regenerating code and stored over a subset of these end users, called storage nodes. This work has studied the performance of the system when end users requests are served either by the base station or by device-to-device communications with the storage nodes.

An important consideration whether the underlying network is wired or wireless is energy efficient operation. To this end, energy harvesting devices offer several advantages such as self-sustainability, perpetual green network operation and increase in energy efficiency. An intense research effort has been devoted in recent years towards understanding the fundamental limits of communications with energy harvesting nodes, especially for wireless networks, see for example [17]–[30]. The salient feature of such systems is the intermittency of instantaneous energy availability, which calls for careful energy management in order to reap the aforementioned advantage. To this end, the problem of minimizing the transmission time for a given number of data packets is introduced in [17]. The short term throughput maximization problem is studied in [18] for the finite battery case and in [19] for fading channels with the assumption of infinitely backlogged data. These approaches have been extended to multi-terminal energy har-

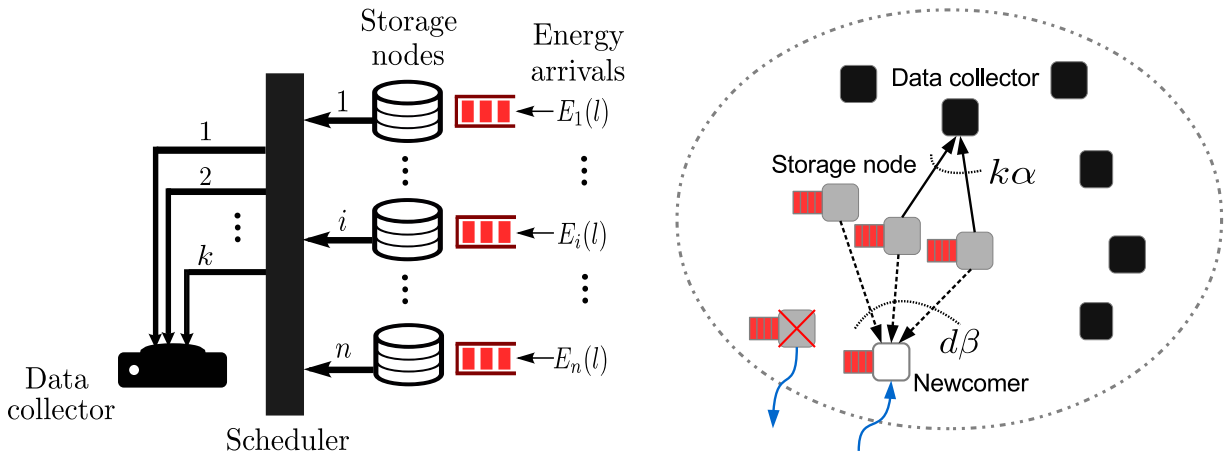
Manuscript received August 1, 2015; revised December 30, 2015; accepted February 23, 2016.

This work was supported in part by the National Science Foundation Grants CCF 14-22347 and CNS 15-26165. This work was presented in part at Asilomar Conference on Signals, Systems, and Computers, Pacific Grove, CA, Nov 2015.

A. M. Ibrahim, A. A. Zewail and A. Yener are with the department of Electrical Engineering, Pennsylvania State University, University Park, PA 16802 USA (e-mail: ami137@psu.edu; aiz103@psu.edu; yener@ee.psu.edu).

Digital Object Identifier

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSAC.2016.2545538, IEEE Journal on Selected Areas in Communications



(a) Data center (b) Wireless sensor network.

Fig. 1. Distributed storage systems with energy harvesting nodes.

vesting networks, nodes energy storage inefficiencies, nodes with processing energy costs, and other battery imperfections [20]–[27]. Long term average throughput optimization with online energy state information has also been studied, see for example, [19], [27]–[29].

Only recently, some initial effort has transpired in investigating the minimization of energy consumption in distributed storage systems. For instance, reference [31] has studied how to store and process data in a k-out-of-n distributed resources system such that the energy consumption is minimized.

Given the prevalence of storing, retrieving, and repairing data in the era of connected networks, it is natural to consider distributed storage systems that are self-energy sustaining, i.e., consisting of energy harvesting devices. Applications of such systems can be wired (see Fig. 1(a)), as in data centers, powered by environmental energy or even wirelessly powering one another; or wireless (see Fig. 1(b)) as have been considered in the recent energy harvesting literature. For data centers, it is known that the electricity costs about 20% of the operational costs [32]. Hence, there is a need for green data centers, where large-scale harvested energy, e.g., solar, can provide a viable solution. For wireless sensor networks, energy harvesting on smaller scale can provide self-sustaining operation [33], [34]. Additionally, as wireless sensor nodes fail often, distributed storage codes can improve reliability [35], [36]. Potential applications include environmental monitoring and infrastructure monitoring, e.g., vibration powered sensors embedded to monitor the structural health of a bridge.

In this work, we consider this new paradigm and study the operational principles of a *distributed storage system with energy harvesting nodes*. The proposed distributed storage system consists of n storage nodes each of which is equipped with an *energy harvester* and a relatively large energy storage device, henceforth called an infinite capacity battery, as in the energy harvesting literature [17], [20], [21], as shown in Fig. 1. F files are stored over the storage nodes using an (n, k, d, α, β) regenerating code. As in a classical distributed storage system, there are retrieval and repair modes. In the retrieval mode, storage nodes serve the files requests of the data collectors, while in the repair mode the alive storage nodes help the newcomer to preserve the system functionality.

First, for a given energy arrival profile for each node, we design transmission algorithms that optimize the system operations, i.e., file retrieval efficiency and node repair time. For the former, we formulate: 1) the problem of maximizing the number of retrieved files by a deadline T, and 2) the problem of minimizing the retrieval time of M different files. The latter, i.e., the repair mode counterparts of these two problems, can be obtained directly by appropriately replacing parameters, as explained in Section III. We derive necessary and sufficient conditions on the energy harvesting profiles that ensure the feasibility of retrieving M files in T time slots. Building on these conditions, we convert the two optimization problems to a single feasibility problem. We then propose algorithms to solve the feasibility problem. Our numerical results demonstrate that, for some system parameters, repairing a failed node by downloading the whole file, using α bits from k alive storage nodes, requires less time on average compared with downloading β bits from d alive storage nodes, where $d\beta < k\alpha$. This demonstrates an operational trade-off between the repair time and energy consumption. Additionally, we extend our formulation to the case where only causal information on the energy arrival profile is available, i.e., the online setup, for maximizing the average number of retrieved files by a deadline T. The online policies are computed numerically evaluating their performance.

The remainder of this paper is organized as follows. In the next section, we describe the system model and the main assumptions. The problem formulations are introduced in Section III. Section IV contains the proposed algorithms for the offline setup. In Section V, we formulate the online version of the problem as a dynamic program. In Section VI, we provide numerical examples on the solutions of the proposed algorithms, evaluating the short and long term performance of the system. Section VII summarizes our conclusions and future directions.

Notation: Throughout the paper, matrices and vectors are represented by boldface uppercase and lowercase letters, re-

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSAC.2016.2545538, IEEE Journal on Selected Areas in Communications

spectively. For a matrix $\boldsymbol{A}$, $A_i(l)$ denotes the element (i,l) and $\boldsymbol{a}(l)$ denotes column l.

II. System Model and Main Assumptions

Consider a data storage system consisting of n nodes, represented by the set $\mathcal{N} = \{1,2,..,n\}$, which store F files, each with size U bits. The files, represented by the set $\mathcal{F} = \{1,2,..,F\}$, are stored over these storage nodes. Each file is encoded using an (n,k,d,α,β) regenerating code [1]–[6], [8]. Each node has data storage capacity $\alpha_T = F\alpha$, and stores an encoded piece of each file.

The distributed data storage system is solely powered by intermittent energy sources, i.e., each storage node is equipped with an energy harvesting unit. Nodes store their harvested energy in infinite capacity batteries, as illustrated in Fig. 1(a). The intermittent energy supply is modeled as a discrete energy harvesting process, where energy arrives at the beginning of each time slot [17]–[19], [30]. Similar to [17]–[19], [30], we initially assume a given energy arrival profile at each node. The system with causal knowledge of energy arrivals is considered in Section V. We denote the energy arrivals to node i at time slot l by $E_i(l)$, which is a sample path from a random process. We consider the energy consumption at the storage nodes due to data transmission. Each node thus has a cost for transmission different from one another, with the thought of capturing channel conditions of each node in a *wireless* scenario, or any circuit or processing costs for either the *wired or wireless* models [26]. To address a variety of applications of a distributed storage system, we consider a general energy cost function $v_i(.)$. A storage node i thus consumes $v_i(b)$ energy units to transmit b bits in one time slot, where $v_i(.)$ is a monotonically increasing function. For simplicity, we consider a time-slotted system with unit length time slots, which allows us to use the terms power and energy interchangeably, as in power allocation versus energy allocation.

The distributed data storage system operates in either one of two modes. The first mode is *retrieval mode*, where a data collector (DC) joins the system to retrieve M files given by the subset of files $\mathcal{M} \subset \mathcal{F}$. The file retrieval process is done file-by-file, such that a data collector retrieves a file when a set of k storage nodes transmit the stored data content related to this file. Furthermore, the transmission occurs only if there exist k storage nodes, each with sufficient energy to transmit α bits. Specifically, a storage node i is capable of transmitting α bits if and only if its stored energy is greater than or equal to $v_i(\alpha)$, which is the energy cost of transmitting α bits from storage node i to the data collector. Communication takes place over one time slot via k orthogonal links. From the received $k\alpha$ bits, the data collector is able to reconstruct the whole file. Similarly, the system repeats this process to retrieve the remaining files in set $\mathcal{M}$. Note that at any time slot during the retrieval mode, we have only two possibilities: if there exist k or more nodes such that each of them has sufficient energy to transmit α bits, then k of those nodes transmit their data to the data collector to retrieve a certain file, otherwise all nodes remain silent. Note that this operational assumption does not affect the number of slots required to retrieve M files, since each node is equipped with infinite capacity battery, and can transmit only one piece of a file in each time slot with fixed energy cost.

The second mode is the *repair mode*, where a newcomer joins the system to replace a failed storage node. Similar to retrieval mode, repair is done file-by-file, such that a file is repaired by the transmission of β bits from d storage nodes over d orthogonal links. A storage node i participates in the repair process if its stored energy is greater than or equal to $v_i(\beta)$, which is the energy cost of communicating β bits from storage node i to the newcomer. With the received $d\beta$ bits, the newcomer generates a piece of size α that preserves the functionality of the system. Again, during the repair mode, the system either repairs the content related to one file by transmitting β bits from d nodes, or all nodes remain silent. Lastly, note that there is no overlap between the two modes of operation. The structure of the distributed storage system and the coding technique [1] impose the aforementioned operational assumptions, which are summarized as follows.

(S1) A file is retrieved (repaired) by transmitting $k\alpha$ ($d\beta$) bits from k (d) nodes, and each node transmits α (β) bits.

(S2) At any time slot, during the retrieval (repair) mode either k (d) nodes transmit or all nodes remain silent.

(S3) The transmission of b bits from storage node i consumes $v_i(b)$ units of energy.

III. Problem Formulation

In this section, we characterize the energy feasibility conditions and formulate two optimization problems that capture the system operations under a given profile of energy arrivals. We focus mainly on the retrieval mode, however, our results can be applied directly to the repair mode for the failure of node i_o, by replacing the parameters $\{M,\mathcal{N},k,\alpha\}$ with $\{F,\mathcal{N}\setminus\{i_o\},d,\beta\}$, respectively. This equivalence can be readily seen from the operational assumptions (S1)-(S3). Therefore, in the remainder of this paper, we present the retrieval mode results and the repair mode counterparts are directly obtained by the aforementioned substitution of parameters, see also Remark 2 in this section for further elaboration on this transformation.

In order to characterize the energy causality conditions on the file retrieval operation, we introduce the notion of *effective accumulated energy*. This notion follows from the assumption that in each time slot, during the retrieval mode, a storage node i either transmits α bits or remains silent. Thus, a storage node i can consume at most $v_i(\alpha)$ units of energy at any time slot.

Definition 1: The **effective accumulated energy** at node i by time slot l, $A_i(l)$, represents the maximum energy that can be utilized at node i by time slot l, and is obtained by the following recursive relation in $z(l-r)$, which represents the maximum energy that can be utilized out of the energy arrivals $E_i(l-r),\ldots,E_i(l)$.

$$A_i(l) = \min\left\{z(2)+E_i(1),\ l\,v_i(\alpha)\right\}, \tag{1}$$

$$z(l-r) = \min\left\{z(l-r+1)+E_i(l-r),\ (r+1)\,v_i(\alpha)\right\}, \tag{2}$$

$$z(l) = \min\left\{E_i(l),\ v_i(\alpha)\right\},\ r=1,\ldots,l-2. \tag{3}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSAC.2016.2545538, IEEE Journal on Selected Areas in Communications

For instance, the effective accumulated energy at node i by time slot 2 is given by

$$A_i(2) = \min\left\{E_i(1) + \min\left\{E_i(2),\ v_i(\alpha)\right\},\ 2\,v_i(\alpha)\right\}. \quad (4)$$

Note that the accumulated energy at node i by time slot j is given by

$$\sum_{l=1}^{j} E_i(l) = A_i(j) + A_i^e(j), \quad (5)$$

where $A_i^e(j)$ is excess energy saved for future use, e.g., $E_i(1) = 4, E_i(2) = 0$, and $v_i(\alpha) = 2$, imply that $A_i(1) = 2, A_i^e(1) = 2, A_i(2) = 4$ and $A_i^e(2) = 0$.

Remark 1: The effective accumulated energy profile of a node can be obtained directly from its energy arrival profile and hence would be available offline as well, if the energy arrival profile is available offline.

Using the notion of effective accumulated energy, we can express the energy causality constraints as

$$\sum_{l=1}^{j} \Delta_i(l) \le \left\lfloor \frac{A_i(j)}{v_i(\alpha)} \right\rfloor \quad \forall\ i \in \mathcal{N}, j = 1, \ldots, T, \quad (6)$$

where $\Delta_i(l) = P_i(l)/v_i(\alpha)$ is the transmission indicator of node i at time slot l and $P_i(l)$ is the transmitted power by node i at time slot l and denotes the element (i, l) in the matrix $\boldsymbol{P}$. Next, we characterize necessary and sufficient conditions on the harvested energy to ascertain the feasibility of retrieving M files by slot T.

Lemma 1: The following conditions for a sequence of time slots $\{l_m\}_{m=1}^{M}$, where $0 < l_1 < \cdots < l_M = T$, are necessary and sufficient for the existence of a feasible power allocation to retrieve M files by T.

$$\sum_{i\in\mathcal{N}} \left\lfloor \frac{A_i(l_m)}{v_i(\alpha)} \right\rfloor \ge km, \quad (7)$$

$$\sum_{i\in\mathcal{N}} \left\lfloor \frac{A_i(l_m)}{v_i(\alpha)} \right\rfloor - \left\lfloor \frac{A_j(l_m)}{v_j(\alpha)} \right\rfloor \ge (k-1)m,$$

$$j = \arg\max\nolimits_{i\in\mathcal{N}}^{(1)} \{A_i(l_m)\}, \quad (8)$$

$$\sum_{i\in\mathcal{S}_{l_m}} \left\lfloor \frac{A_i(l_m)}{v_i(\alpha)} \right\rfloor \ge m,\ \mathcal{S}_{l_m} = \arg\min\nolimits_{i\in\mathcal{N}}^{(n-k+1)} \{A_i(l_m)\}, \quad (9)$$

where $\mathcal{S}_{l_m}$ is the set of $n-k+1$ nodes with minimum effective accumulated energies at time slot l_m, That is, if we have $A_{i_1}(l_m) \le \cdots \le A_{i_{n-k+1}}(l_m) \le \cdots \le A_{i_n}(l_m)$, then, the set $\mathcal{S}_{l_m}$ and its complement $\mathcal{S}_{l_m}^c$ are defined as $\{i_1, \ldots, i_{n-k+1}\}$ and $\{i_{n-k+2}, \ldots, i_n\}$, respectively.

Proof: First, we prove the necessity of the conditions (7)-(9). To this end, we show that the violation of any of the conditions (7)-(9) at time slot T implies that there is no feasible power allocation to retrieve M files by time slot T.

- To prove the necessity of condition (7), suppose

$$\sum_{i=1}^{n} \left\lfloor \frac{A_i(T)}{v_i(\alpha)} \right\rfloor < kM.$$

This simply means that the number of transmissions that could be made by the n nodes till T is fewer than kM, i.e., at least one file will have less than k nodes to contribute to. Thus, at least one file cannot be reconstructed. This contradicts the assumption that we retrieve M files at T.

- Condition (8) ensures that the effective accumulated energy of any node contributes to at most M files. For instance, suppose $\sum_{i=1}^{n} \lfloor A_i(T)/v_i(\alpha) \rfloor = kM$, and $A_{i_o}(T) > M v_{i_o}(\alpha)$, for some $i_o \in \mathcal{N}$, then, we have

$$\sum_{i=1}^{n} \left\lfloor \frac{A_i(T)}{v_i(\alpha)} \right\rfloor - \left\lfloor \frac{A_{i_o}(T)}{v_{i_o}(\alpha)} \right\rfloor < (k-1)M,$$

which implies that node i_o contributes to more than M files. This contradicts the operating assumption that any file can be reconstructed from k distinct nodes. Thus, the system is not able to retrieve M files at T.

- Condition (9) ensures that at least k nodes are able to transmit in M time slots. To see this, suppose $\sum_{i=1}^{n} \lfloor A_i(T)/v_i(\alpha) \rfloor = kM$, and $\sum_{i\in\mathcal{S}_T} \lfloor A_i(T)/v_i(\alpha) \rfloor < M$. This requires the $k-1$ nodes in the set $\mathcal{S}_T^c$ to contribute by more than $(k-1)M$ transmissions, i.e., $\sum_{i\in\mathcal{S}_T^c} \lfloor A_i(T)/v_i(\alpha) \rfloor > (k-1)M$. Then, there is at least one node in the $\mathcal{S}_T^c$ that contributes to more that M transmissions. This contradicts the assumption that to reconstruct a file, k transmissions from k distinct nodes are required. Thus, retrieving M files at T is not possible.

Next, we prove the other direction by showing that if for a sequence of time slots $\{l_m\}_{m=1}^{M}$, where $0 < l_1 < \cdots < l_M = T$, at each l_m the conditions (7)-(9) are satisfied for m files, then there exists a feasible power allocation.

Assume conditions (7)-(9) are satisfied for $m = 1$ at l_1, i.e., $\sum_{i=1}^{n} \lfloor A_i(l_1)/v_i(\alpha) \rfloor \ge k$, $\sum_{i=1}^{n} \lfloor A_i(l_1)/v_i(\alpha) \rfloor - \lfloor A_j(l_1)/v_j(\alpha) \rfloor \ge k-1$, $j = \arg\max_{i\in\mathcal{N}}^{(1)} \{A_i(l_1)\}$ and $\sum_{i\in\mathcal{S}_{l_1}} \lfloor A_i(l_1)/v_i(\alpha) \rfloor \ge 1$. Then, the existence of $k-1$ nodes in the set $\mathcal{S}_{l_1}^c$ with sufficient energy directly follows from (7) and (8). While, (9) ensures the existence of a node in the set $\mathcal{S}_{l_1}$ with sufficient energy to transmit. Consequently, there exists k nodes with sufficient energy and the first file can be retrieved by time slot l_1. Similarly, by applying the same arguments for m files at time slots l_m, we ascertain the feasibility of retrieving M files by time slot T. Note that checking the conditions (7)-(9) at time slot T only is not enough to prove sufficiency due to the fact that a node cannot transmit more than one piece in a time slot. ■

Below we present an example to illustrate the Lemma.

Example 1: Consider the transmission of a file of size $U = 2.4$ under an $(n = 4, k = 3, d = 3, \alpha = 0.8, \beta = 0.8)$ minimum storage regeneration (MSR) code and $v_i(\alpha) = 2,\ \forall\ i$. The effective accumulated energy profiles of the storage nodes are given by

$$\boldsymbol{A} = \begin{bmatrix} 2 & 4 & 4 & 4 \\ 1 & 2 & 4 & 4 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 2 \end{bmatrix}. \quad (10)$$

1) At $l = 1$, none of the conditions (7)-(9) is satisfied.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSAC.2016.2545538, IEEE Journal on Selected Areas in Communications

2) At $l=2$, (7) is satisfied, while (8), (9) are not.

$$\sum_{i=1}^{4}\left\lfloor\frac{A_i(T)}{v_i(\alpha)}\right\rfloor-\left\lfloor\frac{A_1(T)}{v_1(\alpha)}\right\rfloor=1<M(k-1)=2,$$
$$\sum_{i\in\mathcal{S}_2}\left\lfloor\frac{A_i(T)}{v_i(\alpha)}\right\rfloor=0<M,\ \mathcal{S}_2=\{3,4\}.$$

3) At $l=3$, (7), (8) are satisfied, while (9) is not.

$$\sum_{i\in\mathcal{S}_3}\left\lfloor\frac{A_i(T)}{v_i(\alpha)}\right\rfloor=0<M,\ \mathcal{S}_3=\{3,4\}.$$

4) At $l=4$, conditions (7)-(9) are all satisfied.

Therefore, a feasible solution $\boldsymbol{P}$ to retrieve a file by $l=4$, is

$$\boldsymbol{P}=\begin{bmatrix}0&0&0&2\\0&0&0&2\\0&0&0&0\\0&0&0&2\end{bmatrix}. \tag{11}$$

In order to solve the scheduling problem of the file retrieval operation, we formulate the optimization problem of maximizing the number of retrieved files M, given a deadline T, which we refer to as the *throughput maximization* problem. We also consider the problem of minimizing the retrieval time T of M files, which we refer to as the *retrieval time minimization* problem. The throughput maximization problem is given by the following $0-1$ programming problem [37].

$$\textbf{\textit{O1}}:\ \max_{\Delta_i(l)}\quad \frac{1}{k}\sum_{i\in\mathcal{N}}\sum_{l\in\mathcal{T}}\Delta_i(l) \tag{12a}$$
$$\text{subject to}\quad \sum_{l=1}^{j}\Delta_i(l)\le\left\lfloor\frac{A_i(j)}{v_i(\alpha)}\right\rfloor,\ \forall\, i\in\mathcal{N}, j\in\mathcal{T}, \tag{12b}$$
$$\Delta_i(l)\in\{0,1\},\forall\ i\in\mathcal{N},\ l\in\mathcal{T}, \tag{12c}$$
$$\sum_{i\in\mathcal{N}}\Delta_i(l)\in\{0,k\},\ l\in\mathcal{T}, \tag{12d}$$

where $\mathcal{T}=\{1,2,\ldots,T\}$. Here, $\Delta_i(l)$ indicates whether node i transmits in time slot l (1), or not (0). Note that (12b) captures the energy causality constraints, while (12c) and (12d) represent the operational assumptions (S1)-(S3).

Meanwhile, the *retrieval time minimization* problem is given by the following $0-1$ programming problem.

$$\textbf{\textit{O2}}:\ \min_{\Delta_i(l)}\quad T \tag{13a}$$
$$\text{subject to}\quad \sum_{i\in\mathcal{N}}\sum_{l\in\mathcal{T}}\Delta_i(l)=kM, \tag{13b}$$
$$\sum_{l=1}^{j}\Delta_i(l)\le\left\lfloor\frac{A_i(j)}{v_i(\alpha)}\right\rfloor,\ \forall\, i\in\mathcal{N}, j\in\mathcal{T}, \tag{13c}$$
$$\Delta_i(l)\in\{0,1\},\ \forall\ i\in\mathcal{N},\ l\in\mathcal{T}, \tag{13d}$$
$$\sum_{i\in\mathcal{N}}\Delta_i(l)\in\{0,k\},\ l\in\mathcal{T}. \tag{13e}$$

O1 and ***O2*** being binary integer programs, one can pursue methods such as branch and bound [38] to obtain the solution. Here, we will instead use the fact that the optimal solution of a linear integer program can be found in polynomial time if the optimal objective value can be identified in polynomial time [39]. We utilize the structure of the feasible set to develop algorithms that reduce the problems ***O1*** and ***O2*** to a feasibility problem. In particular, the inspection of the energy harvesting profiles, sequentially in time, i.e., utilizing Lemma 1, allows us to find the maximum number of files that can be retrieved in T time slots and the minimum number of time slots needed in the transmission of M files, i.e., identify the optimal objective values for ***O1*** and ***O2***, respectively. It remains then to identify *a feasible* power allocation that can produce these objective values. That is, we have a $0-1$ assignment problem whose solution identifies the transmissions out of each node throughout the session, and thus in turn their transmission power allocation. We term this the *feasibility problem* ***F1***, which can be expressed as

$$\textbf{\textit{F1}}\text{: Find}\quad \boldsymbol{\Delta} \tag{14a}$$
$$\text{subject to}\quad \sum_{i\in\mathcal{N}}\sum_{l\in\mathcal{T}}\Delta_i(l)=kM, \tag{14b}$$
$$\sum_{l=1}^{j}\Delta_i(l)\le\left\lfloor\frac{A_i(j)}{v_i(\alpha)}\right\rfloor,\ \forall\, i\in\mathcal{N}, j\in\mathcal{T}, \tag{14c}$$
$$\Delta_i(l)\in\{0,1\},\ \forall\ i\in\mathcal{N},\ l\in\mathcal{T}, \tag{14d}$$
$$\sum_{i\in\mathcal{N}}\Delta_i(l)\in\{0,k\},\ l\in\mathcal{T}, \tag{14e}$$

where $\Delta_i(l)$ is the (i,l) entry in the matrix $\boldsymbol{\Delta}\in\mathbb{N}^{n\times T}$ which is the transmission indicator matrix of the nodes over T time slots.

Remark 2: A distributed storage system fails if there are fewer than k active nodes. In the case of energy intermittency, the lifetime of a distributed storage system is a concern, since the repair operation is constrained by the harvested energy. Hence, a meaningful objective during the repair mode is minimizing the total time required to repair the F files, i.e., the newcomer completely replaces the failed node. By the substitution of parameters described at the beginning of Section III, i.e., by replacing $\{M,\mathcal{N},k,\alpha\}$ with $\{F,\mathcal{N}\setminus\{i_o\},d,\beta\}$, we obtain the problems that represent the repair mode, i.e., minimizing the repair time of F files, and maximizing the number of repaired files by a deadline T. For example, the repair time minimization problem of node i_o is given by

$$\textbf{\textit{O3}}:\ \min_{\Delta_i(l)}\quad T \tag{15a}$$
$$\text{subject to}\quad \sum_{i\in\mathcal{N}\setminus\{i_o\}}\sum_{l\in\mathcal{T}}\Delta_i(l)=dF, \tag{15b}$$
$$\sum_{l=1}^{j}\Delta_i(l)\le\left\lfloor\frac{\bar{A}_i(j)}{v_i(\beta)}\right\rfloor,\forall\, i\in\mathcal{N}\setminus\{i_o\},\ j\in\mathcal{T}, \tag{15c}$$
$$\Delta_i(l)\in\{0,1\},\forall\, i\in\mathcal{N}\setminus\{i_o\},l\in\mathcal{T}, \tag{15d}$$
$$\sum_{i\in\mathcal{N}\setminus\{i_o\}}\Delta_i(l)\in\{0,d\},\ l\in\mathcal{T}, \tag{15e}$$

where the effective accumulated energy $\bar{A}_i(l)$ is characterized by replacing $v_i(\alpha)$ with $v_i(\beta)$ in (1)-(3). Moreover, the solutions are obtained from the algorithms in Section IV, by the aforementioned parameter substitution.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSAC.2016.2545538, IEEE Journal on Selected Areas in Communications

IV. The proposed Algorithms

In this section, we propose algorithms to solve the aforementioned problems. First, we describe *Algorithms 1* and *2*, provided below, that identify the optimum objective values of ***O1*** and ***O2***.

Algorithm 1 Finds the maximum number of files M to be retrieved by a deadline T.

Input: $A_i(l),\ \forall\ i \in \mathcal{N},\ l \in \mathcal{T}$.
Output: M
1: $M \leftarrow 0$.
2: **for** $l = 1$ **to** T **do**
3: **if** $\sum_{i\in\mathcal{N}} \left\lfloor \frac{A_i(l)}{v_i(\alpha)} \right\rfloor \geq k(M+1)$ **and** $\sum_{i\in\mathcal{N}} \left\lfloor \frac{A_i(l)}{v_i(\alpha)} \right\rfloor - \left\lfloor \frac{A_j(l)}{v_j(\alpha)} \right\rfloor \geq (M+1)(k-1),\ j = \arg\max^{(1)}_{i\in\mathcal{N}}\{A_i(T)\}$ **and** $\sum_{i\in\mathcal{S}_l} \left\lfloor \frac{A_i(l)}{v_i(\alpha)} \right\rfloor \geq (M+1)$ **then**
4: $M \leftarrow M + 1$.
5: **end if**
6: **end for**

Algorithm 2 Finds the minimum retrieval time T of M files.

Input: M, and $A_i(l),\ \forall\ i \in \mathcal{N},\ \forall\ l$.
Output: T
1: $T \leftarrow 0$.
2: **for** $m = 1$ **to** M **do**
3: $T \leftarrow T + 1$.
4: **while** $\sum_{i\in\mathcal{N}} \left\lfloor \frac{A_i(T)}{v_i(\alpha)} \right\rfloor < km$ **or** $\sum_{i\in\mathcal{N}} \left\lfloor \frac{A_i(T)}{v_i(\alpha)} \right\rfloor - \left\lfloor \frac{A_j(T)}{v_j(\alpha)} \right\rfloor < m(k-1),\ j = \arg\max^{(1)}_{i\in\mathcal{N}}\{A_i(T)\}$ **or** $\sum_{i\in\mathcal{S}_T} \left\lfloor \frac{A_i(T)}{v_i(\alpha)} \right\rfloor < m$ **do**
5: $T \leftarrow T + 1$.
6: **end while**
7: **end for**

Observe that the two algorithms simply check the conditions given in Lemma 1 sequentially in time. Also, note that the complexity of *Algorithm 1* is $O(nT)$. Next, we provide two algorithms to find the assignment in ***F1***.

A. Forward algorithm

First, we consider an algorithm that minimizes the average delay per file, by retrieving files as soon as the harvested energy is sufficient and the transmission does not affect the feasibility of transmitting the remaining files. This algorithm is performed over T stages. In stage l, we search for $\boldsymbol{\delta}(l) \in \mathcal{D}_l$, where $\boldsymbol{\delta}(l) = [\Delta_1(l), \ldots, \Delta_n(l)]^T$ and $\mathcal{D}_l$ is the transmissions feasibility set at stage l, defined by

$$\mathcal{D}_l = \Big\{ \boldsymbol{\delta}(l) \in \{0,1\}^n : \Delta_i(l) \leq \left\lfloor \frac{A_i(l)}{v_i(\alpha)} \right\rfloor,\ \forall\ i \in \mathcal{N}, \sum_{i\in\mathcal{N}} \Delta_i(l) \in \{0, k\} \Big\}. \quad (16)$$

To ensure the feasibility of retrieving the remaining files in the upcoming slots, we apply *Algorithm 2* on the updated effective accumulated energies $\hat{A}_i(l),\ i \in \mathcal{N}, l \in \mathcal{T}$ which represent the elements of the matrix $\hat{\boldsymbol{A}}$, as shown in steps 6 to 8.

Algorithm 3 The forward algorithm for finding a feasible power allocation to retrieve M files by time slot T.

Input: T, M and $A_i(l),\ \forall\ i \in \mathcal{N},\ l \in \mathcal{T}$
Output: $\boldsymbol{\Delta}$
1: $\boldsymbol{\Delta} \leftarrow 0$.
2: $f \leftarrow 0$.
3: **while** $l \leq T$ **and** $f < M$ **do**
4: **repeat**
5: Choose $\boldsymbol{\delta}(l) \in \mathcal{D}_l$.
6: $\hat{\boldsymbol{a}}(r) \leftarrow \boldsymbol{a}(r) - [v_1(\alpha)\Delta_1(l), \ldots, v_n(\alpha)\Delta_n(l)]^T,\ r = l, l+1, \ldots, T$.
7: Run *Algorithm 2* with input: $\hat{\boldsymbol{a}}(r),\ r = l, l+1, \ldots, T$.
8: $\hat{M} \leftarrow$ *Algorithm 2* output.
9: **until** $\hat{M} = M - f - 1$.
10: **if** $\boldsymbol{\delta}(l) \neq \mathbf{0}$ **then**
11: Update $\boldsymbol{\Delta}$ with $\boldsymbol{\delta}(l)$.
12: $\boldsymbol{a}(r) \leftarrow \hat{\boldsymbol{a}}(r),\ r = l, l+1, \ldots, T$.
13: $f \leftarrow f + 1$.
14: **end if**
15: $l \leftarrow l + 1$.
16: **end while**

Proposition 1: Algorithm 3 solves the feasibility problem ***F1***.

Proof: At each time slot l, first we choose a transmission strategy $\boldsymbol{\delta}(l)$ from the set of feasible transmission strategies $\mathcal{D}_l$. Second, using *Algorithm 2*, our choice $\boldsymbol{\delta}(l)$ guarantees the sufficiency of remaining energy for transmitting the rest of files. Hence, *Algorithm 3* generates a strategy $\boldsymbol{\Delta}$ that guarantees the transmission of M files in T time slots. ■

Remark 3: Algorithm 3 differs from a myopic approach, where a file is transmitted as soon as a set of k nodes have sufficient energy, such an approach does not guarantee retrieving the requested M files by deadline T. This is illustrated numerically in Subsection VI-A.

Note that the complexity of step 7 is $O(nT)$, steps 4 through 9 are repeated $\binom{n}{k}$ times, and steps 3 through 16 are repeated T times in the worst case. Hence, for a given k the complexity of this algorithm is $O(n\binom{n}{k}T^2)$ and in the worst case is $O(\sqrt{n}\,2^n\,T^2)$.

B. Backward algorithm

In order to avoid searching for a feasible solution at each stage, from 1 to T, (steps 4 to 9 in *Algorithm 3*), we next propose a backward algorithm that has a complexity of $O(nT)$, i.e., more computationally efficient than the forward algorithm. The backward algorithm reduces the number of stages to M stages, $M \leq T$, indexed by q. The stages correspond to time slots T to $T-M+1$, i.e., $l = T - q + 1$.

The first step in the backward algorithm is to capture the energy arrivals $\boldsymbol{E}$ by a transmission opportunity matrix $\boldsymbol{B}$ with $B_i(l)$ as its (i,l) entry. The transmission opportunity matrix $\boldsymbol{B}$ is a binary matrix that shows the time slots at which a node

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSAC.2016.2545538, IEEE Journal on Selected Areas in Communications

is capable of a new transmission. A node i is capable of a new transmission, when its accumulated energy increases by more than or equal to $v_i(\alpha)$ units of energy. Also, we calculate the accumulated transmission opportunity matrix $\boldsymbol{G}$ with $G_i(l)$ as its (i,l) entry. The calculations of $\boldsymbol{B}$ and $\boldsymbol{G}$ are shown in *Algorithm 4*.

Algorithm 4 Finding the transmission opportunity matrices $\boldsymbol{B}$ and $\boldsymbol{G}$.

Input: $\boldsymbol{E}$.
Output: $\boldsymbol{B}$ and $\boldsymbol{G}$.
$\boldsymbol{B} \leftarrow 0$.
for $i = 1$ **to** n **do**
  for $r = 1$ **to** T **do**
    if $\sum_{l=1}^{r} E_i(l) - v_i(\alpha) \left(\sum_{l=1}^{r-1} B_i(l) \right) \geq v_i(\alpha)$ **then**
      $B_i(r) \leftarrow 1$.
    end if
    $G_i(r) \leftarrow \sum_{l=1}^{r} B_i(l)$.
  end for
end for

The following example illustrates the construction of the transmission opportunity matrix $\boldsymbol{B}$ from the energy arrivals matrix $\boldsymbol{E}$.

Example 2: Using *Algorithm 4*, we get the following transmission opportunity matrix, for transmission costs $v_i(\alpha) = i$, $i=1,2,3$, and the energy arrivals matrix $\boldsymbol{E}$.

$$\boldsymbol{E} = \begin{bmatrix} 0 & 0 & 0 & 2 & 0 & 2 & 1 & 1 & 1 & 0 \\ 0 & 1 & 2 & 0 & 2 & 1 & 1 & 0 & 2 & 1 \\ 1 & 2 & 1 & 0 & 0 & 2 & 0 & 0 & 0 & 0 \end{bmatrix}, \tag{17}$$

$$\boldsymbol{B} = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{18}$$

In the backward algorithm, we schedule the nodes with most recent transmission opportunity. Particularly, at stage q, we give the priority to the set of nodes with current transmission opportunity, $\mathcal{B}_q$, which is defined as

$$\mathcal{B}_q = \{ i \in \mathcal{N} : \ B_i(T-q+1) = 1 \}, \tag{19}$$

and its complement is denoted by $\mathcal{B}_q^c$. If the number of nodes in the set $\mathcal{B}_q$ is less than k, we schedule the rest starting with the nodes that have maximum accumulated energy. For instance, if the set $\mathcal{B}_q$ has y nodes, we schedule the remaining $k-y$ nodes from the set $\mathcal{C}_q$ given by

$$\mathcal{C}_q = \arg\max_{i \in \mathcal{B}_q^c}^{(k-y)} \{ G_i(T-q+1) \}. \tag{20}$$

Finally, we update the transmission opportunity matrix $\boldsymbol{B}$ and the accumulated transmission opportunity matrix $\boldsymbol{G}$, by subtracting the allocated transmissions. The steps of the backward scheduling algorithm are illustrated in *Algorithm 5*.

Algorithm 5 The backward algorithm for finding a feasible power allocation to retrieve M files by time slot T.

Input: T, M, and $\boldsymbol{E}$.
Output: $\boldsymbol{\Delta}$.
$\boldsymbol{\Delta} \leftarrow 0$.
for $q = 1$ **to** M **do**
  if $\sum_{i \in \mathcal{N}} B_i(T-q+1) \geq k$ **then**
    $\Delta_{i_v}(T-q+1) \leftarrow 1$ for some $\{i_1, \ldots, i_k\} \subseteq \mathcal{B}_q$
  else
    $\Delta_i(T-q+1) \leftarrow 1, \forall\ i \in \mathcal{B}_q$.
    $y \leftarrow \sum_{i \in \mathcal{N}} B_i(T-q+1)$.
    $\mathcal{C}_q \leftarrow \arg\max_{i \in \mathcal{B}_q^c}^{(k-y)} \{ G_i(T-q+1) \}$.
    $\Delta_i(T-q+1) \leftarrow 1, \forall\ i \in \mathcal{C}_q$.
  end if
  for $i = 1$ **to** n **do**
    $r \leftarrow q$,
    $w_i \leftarrow \Delta_i(T-q+1)$.
    while $w_i > 0$ **do**
      if $B_i(T-r+1) > 0$ **then**
        $B_i(T-r+1) \leftarrow 0, \ \ w_i \leftarrow 0$.
      end if
      $r \leftarrow r+1$.
    end while
  end for
  Update $\boldsymbol{G}$ with $\boldsymbol{B}$.
end for

Proposition 2: Algorithms 4 and 5 solve the feasibility problem ***F1***.

Proof: The proof is by construction. Each node is equipped with infinite capacity battery, hence it is sufficient to consider the last M time slots for retrieving the required M files. First, consider the case where conditions (7)-(9) holds with equality, i.e., there is no excess energy. A transmission opportunity at time slot $T-M+m$ is essential for the transmission of file m, since there is no excess energy and the algorithm schedules the files in the last M time slots. Hence, at stage q, we must schedule the nodes in the set $\mathcal{B}_q$. If the number of nodes in $\mathcal{B}_q$ is $y < k$, then for the remaining $k-y$ nodes we have the following two cases.

- The set $\mathcal{B}_q$ does not include any of the nodes in the set $\mathcal{S}_q$, i.e., the number of nodes in the set $\mathcal{B}_q^c \cap \mathcal{S}_q^c$ is $k-y-1$. This implies that the set $\mathcal{C}_q$ contains one node from the set $\mathcal{S}_q$.
- The set $\mathcal{B}_q$ includes one node from the set $\mathcal{S}_q$, i.e., the number of nodes in the set $\mathcal{B}_q^c \cap \mathcal{S}_q^c$ is $k-y$. This implies that $\mathcal{C}_q = \mathcal{B}_q^c \cap \mathcal{S}_q^c$.

In both cases, scheduling the $k-y$ nodes in the set $\mathcal{C}_q$ implies that the algorithms have scheduled $k-1$ nodes from the set $\mathcal{S}_q^c$ and one node from the set $\mathcal{S}_q$. This agrees with the sufficiency proof in Lemma 1.

Next, consider the case where one of the conditions (7)-(9) holds with strict inequality, i.e., there is excess energy. In addition to the previous two cases when the number of nodes in $\mathcal{B}_q$ is $y < k$, we have the following case.

- The set $\mathcal{B}_q$ includes $y_{\mathcal{S}} > 1$ nodes from the set $\mathcal{S}_q$. However, the $y_{\mathcal{S}} - 1$ excess transmission opportunity at stage q, are due to the excess energy.

Hence, scheduling the $k-y$ nodes in the set $\mathcal{C}_q$ guarantees the feasibility of scheduling the remaining files. ■

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSAC.2016.2545538, IEEE Journal on Selected Areas in Communications

V. Online algorithms

The scheduling policies proposed in the previous sections require offline knowledge of the energy arrivals, i.e., the energy harvesting profiles over the operation duration are known in advance. In this section, we investigate the online version of the problem of maximizing the average number of retrieved files by a deadline T, under causal knowledge of energy arrivals, i.e., the scheduling decision at each time slot depends only on the previous decisions and energy arrivals. The optimal online policy can be found using dynamic programming [40], under the assumption that energy arrivals are i.i.d. over time slots.

Define the state of the system at time slot l as the accumulated energy in the batteries, denoted by $\boldsymbol{x}(l) = [X_1(l), \dots, X_n(l)]^T$. The action at time slot l is the transmitted powers, $\boldsymbol{p}(l) \in \mathcal{P}_l(\boldsymbol{x}(l))$, where

$$\mathcal{P}_l(\boldsymbol{x}(l)) = \Big\{\boldsymbol{p}(l) : P_i(l) \in \{0, v_i(\alpha)\},\ P_i(l) \le X_i(l), \\ \forall i \in \mathcal{N},\ \sum_{i=1}^{n} \frac{P_i(l)}{v_i(\alpha)} \in \{0, k\}\Big\}. \quad (21)$$

Hence, the state of the system, $\boldsymbol{x}(l)$, evolves as follows

$$\boldsymbol{x}(l+1) = \boldsymbol{x}(l) + \boldsymbol{e}(l+1) - \boldsymbol{p}(l), \quad (22)$$

where $\boldsymbol{e}(l) = [E_1(l), \dots, E_n(l)]^T$ denotes the energy arrivals at time slot l. Note that the state of the system evolves according to the probability distribution

$$\mathbb{P}\big[\ \boldsymbol{x}(l+1) = \boldsymbol{x} \mid \boldsymbol{x}(l),\ \boldsymbol{p}(l)\ \big] = \\ \mathbb{P}\big[\ \boldsymbol{e}(l+1) = \boldsymbol{x} - \boldsymbol{x}(l) + \boldsymbol{p}(l) \,|\, \boldsymbol{x}(l),\ \boldsymbol{p}(l)\ \big]. \quad (23)$$

An online policy is defined by $\pi = \{\mu_1, \dots, \mu_T\}$, where μ_l is a mapping between the state $\boldsymbol{x}(l)$ and the action $\boldsymbol{p}(l)$, i.e., $\boldsymbol{p}(l) = \mu_l(\boldsymbol{x}(l))$ and $\mu_l(\boldsymbol{x}(l)) = \left[\mu_l^1(X_1(l)), \dots, \mu_l^n(X_n(l))\right]^T$. We define the reward at time slot l as the number of retrieved files. Formally, the reward function at time slot l is given by

$$g_l\big(\mu_l(\boldsymbol{x}(l))\big) = \frac{1}{k} \sum_{i=1}^{n} \frac{\mu_l^i(X_i(l))}{v_i(\alpha)},\ l \in \mathcal{T}. \quad (24)$$

Given an initial state $\boldsymbol{x}(1) = \boldsymbol{e}(1)$, the expected number of retrieved files by time slot T under policy π is given by

$$J_\pi(\boldsymbol{x}(1)) = \mathbb{E}\left[\sum_{l=1}^{T} g_l\big(\mu_l(\boldsymbol{x}(l))\big)\right]. \quad (25)$$

Hence, the optimal value function, given an initial state $\boldsymbol{x}(1)$, is expressed as

$$J_{\pi^*}(\boldsymbol{x}(1)) = \max_{\pi}\ J_\pi(\boldsymbol{x}(1)), \quad (26)$$

where π^* is the optimal online policy. This optimal policy is obtained by solving Bellman's equation which is given by

$$J_l\big(\boldsymbol{x}(l)\big) = \max_{\boldsymbol{p}(l) \in \mathcal{P}_l(\boldsymbol{x}(l))} g_l\big(\mu_l(\boldsymbol{x}(l))\big) + \mathbb{E}\big[J_{l+1}\big(\boldsymbol{x}(l+1)\big)\big], \quad (27)$$

$$= \max_{\boldsymbol{p}(l) \in \mathcal{P}_l(\boldsymbol{x}(l))} g_l\big(\mu_l(\boldsymbol{x}(l))\big) + \sum_{x \in \mathcal{X}} J_{l+1}\big(\boldsymbol{x}(l+1)\big) \\ \times \mathbb{P}\big[\boldsymbol{x}(l+1) = \boldsymbol{x} | \boldsymbol{x}(l), \boldsymbol{p}(l)\big], \quad (28)$$

where $\mathcal{X}$ describes the set of all possible states. The solution of Bellman's equation can be obtained using value iteration [40].

VI. Numerical results and discussion

In this section, we present numerical results to demonstrate the performance of a distributed storage system with energy intermittency in different settings and discuss the resulting insights. Each subsection describes the scenarios we have evaluated and compared.

A. Myopic vs Optimal Policies

In this subsection, we first present an example that demonstrates the solutions obtained by the proposed algorithms and a myopic one. In the myopic algorithm, the system retrieves a file once a set of k nodes with sufficient energy exists, if there are more than k nodes with sufficient energy, it randomly chooses any k of them, regardless of the feasibility of retrieving the remaining files by the deadline T.

Example 3: Consider a distributed storage system consisting of $n = 4$ nodes and operates at the MSR point. A file, with size 2.4 Mbits, is retrieved from any $k = 3$ nodes each transmitting $\alpha = 0.8$ Mbits. We assume symmetric transmission costs $v_i(\alpha) = 2,\ \forall i$. Our objective is to retrieve $M = 5$ files by a deadline $T = 9$. For the effective accumulated energy given by

$$\boldsymbol{A} = \begin{bmatrix} 0 & 0 & 1 & 2 & 3 & 5 & 5 & 6 & 7 \\ 0 & 1 & 1 & 2 & 4 & 4 & 4 & 4 & 6 \\ 0 & 1 & 3 & 4 & 5 & 5 & 5 & 7 & 9 \\ 2 & 3 & 5 & 7 & 7 & 7 & 8 & 10 & 10 \end{bmatrix}, \quad (29)$$

we obtain the following solutions by the forward and backward algorithms, respectively

$$\boldsymbol{P}_F = \begin{bmatrix} 0 & 0 & 0 & 0 & 2 & 2 & 0 & 2 & 0 \\ 0 & 0 & 0 & 2 & 0 & 2 & 0 & 0 & 2 \\ 0 & 0 & 0 & 2 & 2 & 0 & 0 & 2 & 2 \\ 0 & 0 & 0 & 2 & 2 & 2 & 0 & 2 & 2 \end{bmatrix}, \quad (30)$$

$$\boldsymbol{P}_B = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 2 & 2 & 2 & 0 \\ 0 & 0 & 0 & 0 & 2 & 0 & 2 & 0 & 2 \\ 0 & 0 & 0 & 0 & 2 & 2 & 0 & 2 & 2 \\ 0 & 0 & 0 & 0 & 2 & 2 & 2 & 2 & 2 \end{bmatrix}, \quad (31)$$

while the myopic algorithm yields

$$\boldsymbol{P}_{\text{myopic}} = \begin{bmatrix} 0 & 0 & 0 & 2 & 0 & 0 & 0 & 2 & 2 \\ 0 & 0 & 0 & 2 & 2 & 0 & 0 & 0 & 2 \\ 0 & 0 & 0 & 2 & 2 & 0 & 0 & 2 & 2 \\ 0 & 0 & 0 & 0 & 2 & 0 & 0 & 2 & 0 \end{bmatrix}. \quad (32)$$

One can observe that in both solutions, $\boldsymbol{P}_F$ and $\boldsymbol{P}_{myopic}$ the system starts retrieving files at the fourth time slot. The optimal algorithm is capable of retrieving the $M = 5$ files by the deadline T, while the myopic algorithm retrieves only 4 files. Note that each of the solutions $\boldsymbol{P}_F$ and $\boldsymbol{P}_B$ retrieves the 5 required files, however, $\boldsymbol{P}_F$ is better in sense of average delay per file, at the expense of having a higher complexity. In Fig. 2, we compare the average number of files retrieved by the optimal policy in T time slots with the myopic policy.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSAC.2016.2545538, IEEE Journal on Selected Areas in Communications

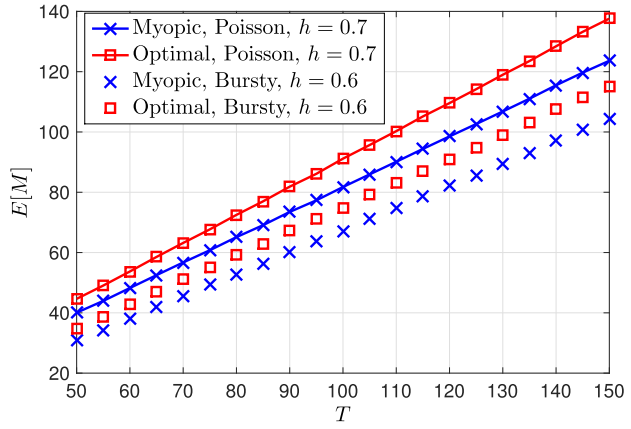


Fig. 2. Comparing the optimal offline and myopic approaches for $n = 7$, $k = 4$, $\alpha = 0.5$ Mbits, $\beta = 1/6$ Mbits, $d = 6$, file size $U = 2$ Mbits, $v_i(\alpha) = 1$, for $i = 1, 4, 6, 7$, $v_i(\alpha) = 2$ for $i = 2, 5$, and $v_3(\alpha) = 3$.

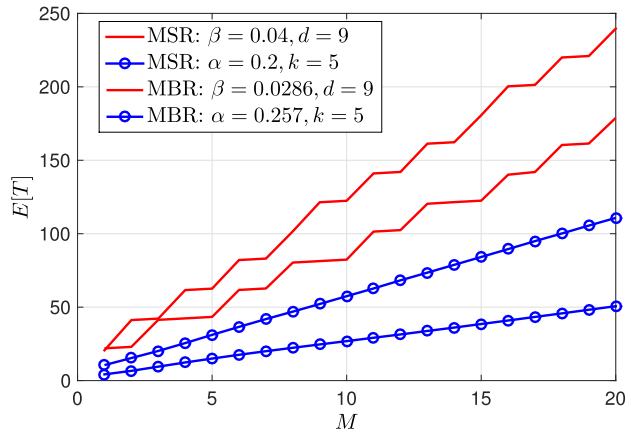


Fig. 3. The average minimum file retrieval (repair) time $\mathbb{E}[T]$ for M files when $h_i = 0.1, i = 1, ..., 8, 10$ and $h_9 = 0.01$.

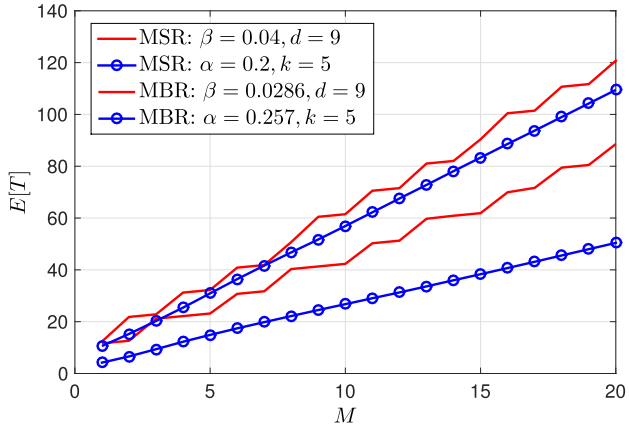


Fig. 4. The average minimum file retrieval (repair) time $\mathbb{E}[T]$ for M files when $h_i = 0.1, i = 1, ..., 8, 10$ and $h_9 = 0.02$.

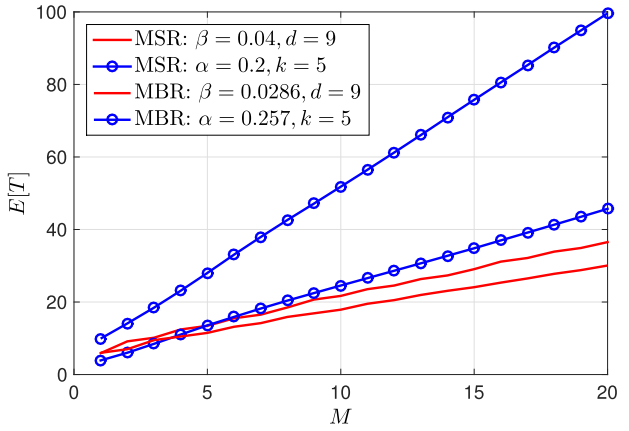


Fig. 5. The average minimum file retrieval (repair) time $\mathbb{E}[T]$ for M files when $h_i = 0.1, i = 1, ..., 9, 10$.

The results demonstrate the performance gap between the optimal and myopic policies under different energy arrivals. In particular, we consider i.i.d. $E_i(l) \sim \text{Poisson}(h)$, $\forall i$, and i.i.d. bursty energy arrivals with burst size $\theta = 5$ and harvesting rate h, i.e., $E_i(l) = \theta$ with probability h/θ and $E_i(l) = 0$ with probability $1 - h/\theta$.

B. Minimum Storage Regenerating (MSR) vs Minimum Bandwidth Regenerating (MBR) codes

We compare the performance of a system storing M files, each with size 1 Mbits, at the two extreme points on the storage and repair bandwidth trade-off curve [1], i.e., the MSR and MBR codes. We consider $(n = 10, k = 5, d = 9, \alpha, \beta)$ codes: 1) For MSR, $\alpha = 0.2$ Mbits and $\beta = 0.04$ Mbits; and 2) For MBR, $\alpha = 0.257$ Mbits and $\beta = 0.0286$ Mbits. The transmission cost is given by $v_i(b) = 2^{2b} - 1$ $\forall i$, where b is in Mbits. We investigate different cases for the energy harvesting rates, where the energy arrivals at each node follow an i.i.d Binomial distribution with maximum arrival $\theta = 0.5$ energy units, i.e., $10 \times E_i(t) \sim B(5, h_i)$, $\forall i$.

Fig. 3, 4, and 5 show the average minimum file retrieval (repair) time for M files under different energy harvesting rates. From Fig. 3, we observe that retrieving the whole file requires less time on average compared with the repair process at the two extreme points, MSR and MBR. On the other hand, Fig. 4 shows that retrieving the whole file requires less time at the MSR point, while the repair process requires less time at the MBR point. Fig. 5 indicates that the repair process requires less time for both MSR and MBR when the number of files in the system is large enough, in this example $M > 5$.

From the aforementioned observations, we conclude that the repair time is highly dependent on the energy arrivals. In some cases, coding schemes that achieve minimum repair bandwidth may lead to delay in the repair process under intermittency of energy. In these cases, repair by reconstructing the whole file may be more delay efficient compared to conventional repair, which is energy efficient.

C. Online vs Offline

Fig. 6 shows the performance of the system under causal knowledge of energy arrivals (online) and non-causal knowl-

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSAC.2016.2545538, IEEE Journal on Selected Areas in Communications

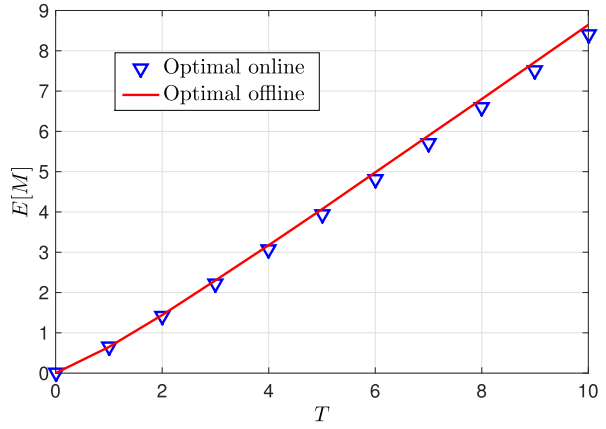


Fig. 6. Comparing the optimal offline and online policies for $n=4$, $k=d=3$, $\alpha=\beta=0.5$ Mbits, $U=1.5$ Mbits, $v_i(\alpha)=1\ \forall i$ and i.i.d Bernoulli energy arrivals with mean $h=0.7$.

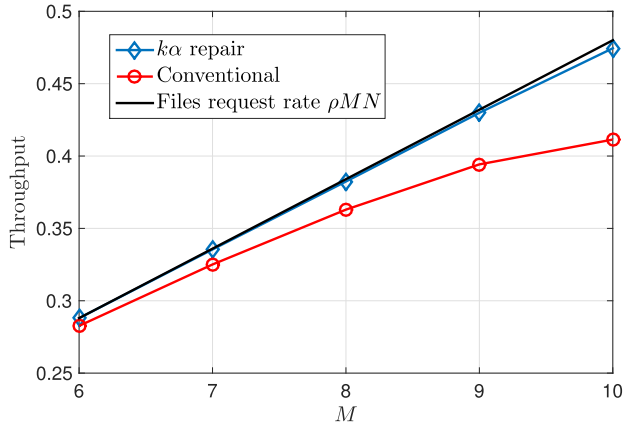


Fig. 7. Comparing the throughput of a conventional system with the throughput of a system implementing the $k\alpha$ repair strategy.

edge of energy arrivals (offline). We compare the average number of retrieved files in T time slots under the optimal online policy proposed in Section V, and the optimal offline algorithms in Section IV. From Fig. 6, we observe that, for small T, the gap between the offline and online solutions is small, so the offline solution can serve as a good indicator of performance.

D. Multi-epoch simulations

In order to evaluate the multi-epoch behavior of a distributed storage system with energy harvesting nodes, we consider a system consisting of N nodes and stores M files over n of them, where $n << N$. A similar set up was investigated in [15]. The N nodes send file retrieval requests to the n storage nodes, which are stored in a file requests queue. We model the file requests as a Poisson process with rate equals to ρMN, where ρ captures the intensity of file requests. The system dynamics change periodically every T time slots, i.e., at the beginning of each epoch, we know the energy harvesting profiles and the transmission costs during this epoch. At the end of an epoch, a storage node leaves the system with probability u and a newcomer replace it, see Fig. 1(b). We need to initiate a repair process whenever a storage node leaves the system, which happens with probability $1-(1-u)^n$. We assume that the repair process is given the priority over the file retrieval process, i.e., the storage nodes start serving the file retrieval requests after repairing all the failed nodes. We focus on the case $M \leq T$, since for $M > T$ the system is overloaded and the repair process needs at least two epochs.

Next, we investigate the key performance metrics for the aforementioned distributed storage system. In particular, we evaluate the average fraction of time slots in which the storage nodes are serving the file requests, which we refer to as the system *throughput*. Additionally, we calculate the average fraction of time slots in which the system is in the repair mode, which we refer to as the system *repair rate*. Intuitively, the system throughput is upper bounded by the file requests rate and the system repair rate is lower bounded by $M(1-(1-u)^n)$,

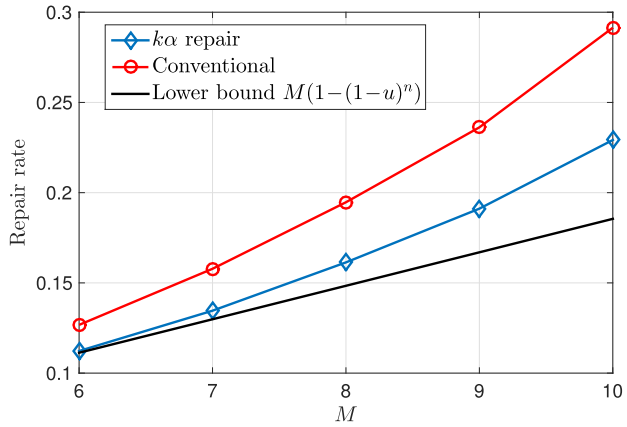


Fig. 8. Comparing the repair rate of a conventional system with the repair rate of a system implementing the $k\alpha$ repair strategy.

since each file needs one time slot to be repaired. Also, we define the *lifetime* of the system as the average number of epochs beyond which the number of active storage nodes is less than k.

In Fig. 7, 8 and 9, we show the throughput, repair rate and lifetime of a system employing an $(n=4, k=2, d=3, \alpha=0.6, \beta=0.2)$ MBR code, with file size $U=1$ Mbits. We assume an epoch length $T=10$, file requests intensity $\rho=0.12/N$ and the probability that a storage node leaves the system $u=0.05$. We assume a Poisson energy harvesting process of rate $h_i=0.5$, $\forall i$ and fixed transmission costs $v_i(\alpha)=1.3$, $v_i(\beta)=0.32$, $\forall\ i$. We also study a system in which a file is repaired by downloading α bits from k storage nodes, i.e., the whole file is reconstructed in the repair process. In Fig. 7, 8 and 9, we compare the performance of a distributed storage system implementing the $k\alpha$ repair strategy with the performance of the conventional system, where a file is repaired by downloading β bits from d nodes. From Fig. 7, we observe that the $k\alpha$ repair strategy enhances the system throughput significantly. Additionally, it decreases the repair rate and increases the lifetime of the system, as shown in Fig.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSAC.2016.2545538, IEEE Journal on Selected Areas in Communications

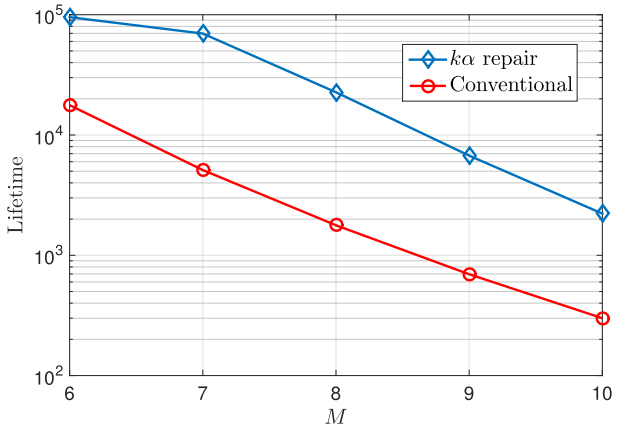


Fig. 9. Comparing the lifetime of a conventional system with the lifetime of a system implementing the $k\alpha$ repair strategy.

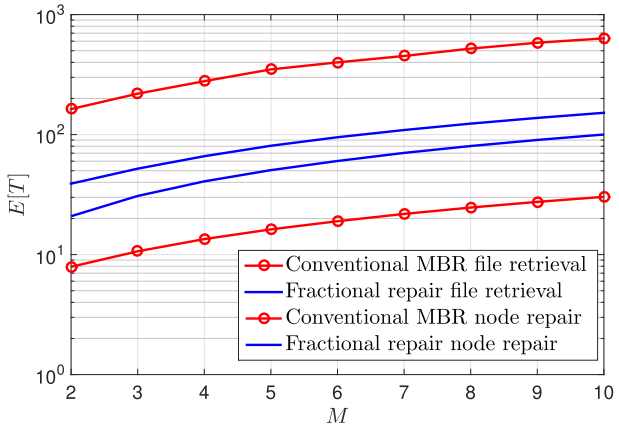


Fig. 10. Comparing a fractional repair code [5] and a MBR code with random access repair for $(n = 6, k = 3, d = 3, \beta = 1, \alpha = 3)$. $E_i(l) \sim$ Poisson(h_i), $h_i = 0.8$ for $i = 1, 2.., 5$ and $h_6 = 0.3$.

8 and 9.

E. Extensions and future directions

In this subsection, we investigate several extensions of the distributed storage system with energy harvesting nodes under different coding criteria.

1) Fractional repair codes: In the coding scheme proposed in [5] for MBR, downloading at most $\alpha-1$ data symbols, from any k nodes, is sufficient for reconstructing the file. The proper choice of the transmitted symbols, prevents downloading redundant bits. This approach clearly leads to energy savings and can decrease the average retrieval time as demonstrated in Fig. 10. The solution for file retrieval in this case is obtained by replacing α with $\alpha - 1$ in our algorithms. We note that the repair process under this coding technique is table-based, i.e., to regenerate the content of a failed node, the newcomer has to connect a specific set of d nodes. Under the energy intermittency, this restriction may cause delays in the repair process as evident from Fig. 10. In the case where each data symbol has two replica stored in two different nodes, our algorithms are applicable in the repair mode by restricting the set of nodes for which we check the necessary and sufficient conditions to the set defined by the repair table. Note that if there are more than two replicas for each data symbol, the set of choices in the repair process is enlarged, however the random access repair time is still lower. As a last comment, we note that, in the retrieval mode, we have only considered the case where each node transmits $\alpha - 1$ bits. In general, nodes can transmit varying number of bits. Treatment of this is left as a future direction.

2) Non-homogeneous systems: In this work, we have considered systems with similarly capable nodes, i.e., a homogeneous system. Our methodology and solutions, however, are applicable to non-homogeneous distributed storage systems studied in references [41] and [42], as well.

In [41], one of the storage nodes is considered a super node that has double the capacity of any of the remaining nodes. In case a regular node fails, the super node steps in and transmits double the bits transmitted by each of the regular nodes. In case of the failure of two regular nodes, or the super node, the repair is done by reconstructing the whole file. In summary, whenever the super node participates in the repair process, it transmits double the bits transmitted by a regular node. Consequently, we can apply our algorithms to this setup, by replacing β with 2β in the cost function of the super node. We investigate the performance of this non-homogeneous system, by comparing it with a homogeneous system that has the same total storage capacity and total energy harvesting rate. In particular, we consider a non-homogeneous system consisting of 4 nodes, where the first node is a super node that has storage capacity $2\alpha_T$, and energy harvesting rate h_{super}, while each of the regular nodes has storage capacity α_T and harvesting rate h_{ri}, where $i = 2, 3, 4$. On the other hand, the homogeneous system consists of 5 nodes each of which has storage capacity α_T and harvesting rate h_i, where $i = 1, .., 5$. To have a fair comparison, we assume that $h_{ri} = h_{i+1}, i = 2, 3, 4$, and $h_{\text{super}} = h_1 + h_2$. In both systems, we assume that node 4 fails and all active nodes participate in the repair process. Each regular node transmits β bits to the newcomer, while the super node transmits 2β bits. The transmission cost is given by $v_i(b) = 2^{2b} - 1\ \forall i$, where b is in Mbits. For a file size $U = 6$ Mbits, Fig. 11 shows that the non-homogeneous system achieves lower average repair time compared with the homogeneous system with asymmetric energy arrivals. The homogeneous system performs better under symmetric energy arrivals. On the other hand, for a file size $U = 3$ Mbits, the non-homogeneous system always outperforms the homogeneous one as evident from Fig. 12.

Similarly, modifying the cost function during the repair process captures the nature of the two-rack non-homogeneous distributed storage model considered in [42]. In this model, the repair bandwidth differs depending on the rack of the failed node and the nodes involved in the repair process. If a node participates in repairing another node in the same rack, then the repair bandwidth is given by β bits. On the other hand, when a node participates in repairing a node in the other rack, then the repair bandwidth is given by $\tau\beta$ bits, where $\tau > 1$.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSAC.2016.2545538, IEEE Journal on Selected Areas in Communications

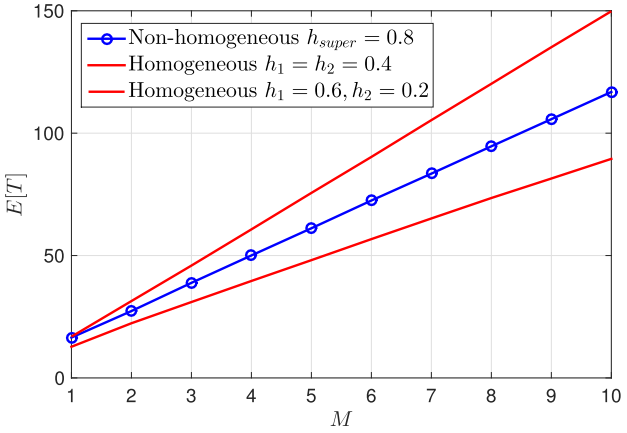


Fig. 11. Comparing the homogenous and non-homogenous systems, under different harvesting rates, for $\beta = 1$ Mbits, and $U = 6$ Mbits.

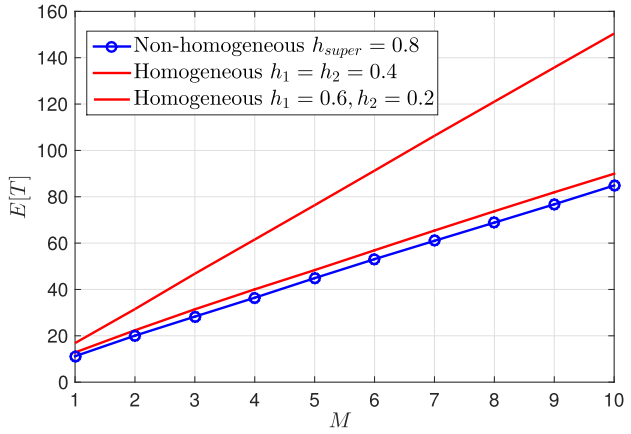


Fig. 12. Comparing the homogenous and non-homogenous systems for $\beta = 0.5$, and $U = 3$ Mbits.

During the repair process of failed node in the first rack, we modify the cost functions of the nodes located in the second rack by replacing β with $\tau\beta$.

3) Local repair: Recently, locally repairable codes have gained a lot of interest, see for example, [43], [44]. In these coding schemes, the storage nodes are divided into groups, such that in case a node fails, its neighbors in the same group, can initiate the repair process. This in turn reduces the repair bandwidth. We consider a system consisting of 12 nodes that utilizes the coding scheme described in Fig. 1 in reference [44]. The nodes are divided into 3 groups, each containing 4 nodes. In case of node failure, either two nodes from its group perform a local repair, or the whole file is retrieved by the newcomer. We assume local repair cost $v_i(\beta) = 3$, $\forall\, i$ and file retrieval cost $v_i(\alpha) = 4$, $\forall\, i$. Fig. 13 shows the average repair time of a single file by the two methods. The energy arrivals, at each node in group i, are i.i.d. Poisson realizations with rate h_i. As expected, as the energy harvesting rate increases the average local repair time decreases. We observe that, for group 3, retrieving the whole file is more time efficient than local repair. Our proposed algorithms can be applied to the

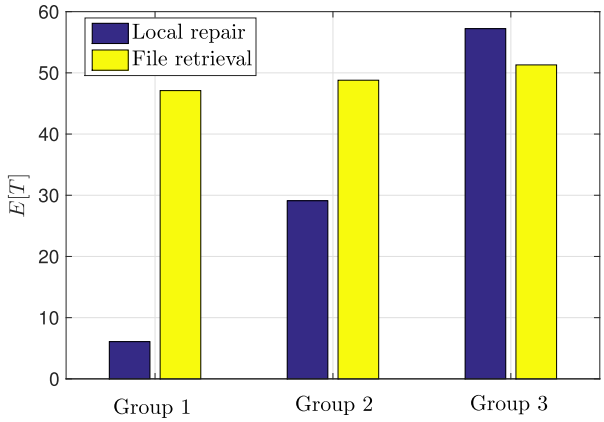


Fig. 13. Comparing the average repair time of one file for $h_1 = 0.5$, $h_2 = 0.2$, and $h_3 = 0.1$.

local repair mode by restricting the set of nodes, for which we check the necessary and sufficient conditions, to the set of alive nodes in the same group. On the other hand, the file retrieval process requires new energy feasibility conditions. For this example, a data collector should connect to $k = 5$ different nodes, such that it connects to at least one node from each group and at most two nodes from the same group. This leads to the notion of intra-group and inter-group energy feasibility conditions for the file retrieval process. Deriving these conditions for *multiple files* is left as a future direction.

VII. Conclusions

In this paper, we have investigated the effect of intermittency of energy availability, i.e., energy harvesting constraints, on a distributed storage system with multiple files. We have formulated two optimization problems that represent the file retrieval and node repair processes. In order to solve these optimization problems, first we have characterized the necessary and sufficient energy feasibility conditions to retrieve M files in T time slots. Second, using these conditions, we have developed two algorithms that reduce the optimization problems to a feasibility problem. Third, we have solved this feasibility problem using forward and backward algorithms and demonstrated their solutions by numerical examples. The forward algorithm minimizes the average delay per file, while the backward algorithm is more computationally efficient as it solves an equivalent binary problem. In addition, we have investigated the optimal online policy for the system under causal knowledge of energy arrivals. Finally, we have provided numerical results that demonstrates the performance of the system under several parameters values. The results indicate that, with energy harvesting nodes, codes that minimize the repair bandwidth may not lead to minimum repair time. Thus, an operational trade-off arises between the repair time and energy consumption. Additionally, the choice of the number of stored files M differs depending on the system objectives such as lifetime, repair rate and throughput.

This work is an initial attempt to investigate the dynamics of distributed storage systems under the intermittently available

harvested energy. Future directions to consider include general heterogeneous systems, e.g., files with different sizes or different coding parameters; distributed storage systems with finite capacity batteries; and energy cooperation between the system nodes.

REFERENCES

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

[2] K. Rashmi, N. B. Shah, P. V. Kumar, and K. Ramchandran, "Explicit construction of optimal exact regenerating codes for distributed storage," in *Proc. IEEE 47th Annu. Allerton Conf. Commun. Control Comput.*, Sep. 2009, pp. 1243–1249.

[3] C. Suh and K. Ramchandran, "Exact-repair MDS codes for distributed storage using interference alignment," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jun. 2010, pp. 161–165.

[4] K. V. Rashmi, N. B. Shah, and P. V. Kumar, "Optimal exact-regenerating codes for distributed storage at the MSR and MBR points via a product-matrix construction," *IEEE Trans. Inf. Theory*, vol. 57, no. 8, pp. 5227–5239, Aug. 2011.

[5] S. El Rouayheb and K. Ramchandran, "Fractional repetition codes for repair in distributed storage systems," in *Proc. IEEE 48th Annu. Allerton Conf. Commun. Control Comput.*, Sep. 2010, pp. 1510–1517.

[6] 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, May 2013.

[7] S. Mohajer and R. Tandon, "New bounds on the (n, k, d) storage systems with exact repair," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jun. 2015, pp. 2056–2060.

[8] A. G. Dimakis, K. Ramchandran, Y. Wu, and C. Suh, "A survey on network codes for distributed storage," *Proc. IEEE*, vol. 99, no. 3, pp. 476–489, Mar. 2011.

[9] M. Gerami and M. Xiao, "Repair for distributed storage systems with erasure channels," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2013, pp. 4058–4062.

[10] M. Gerami, M. Xiao, and M. Skoglund, "Optimal-cost repair in multi-hop distributed storage systems," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jul. 2011, pp. 1437–1441.

[11] S. Chen, Y. Sun, U. C. Kozat, L. Huang, P. Sinha, G. Liang, X. Liu, and N. B. Shroff, "When queueing meets coding: Optimal-latency data retrieving scheme in storage clouds," in *Proc. IEEE INFOCOM*, Apr. 2014, pp. 1042–1050.

[12] N. Golrezaei, A. G. Dimakis, and A. F. Molisch, "Device-to-device collaboration through distributed storage," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2012, pp. 2397–2402.

[13] J. Paakkonen, C. Hollanti, and O. Tirkkonen, "Device-to-device data storage for mobile cellular systems," in *Proc. IEEE Global Commun. Conf. Globecom Workshops*, Dec. 2013, pp. 671–676.

[14] D. Karpuk, C. Hollanti, and A. Barreal, "Node repair for distributed storage systems over fading channels," in *Proc. IEEE Int. Sym. Inf. Theory Applicat. (ISITA)*, Oct. 2014, pp. 383–387.

[15] J. Pedersen, A. Graell i Amat, I. Andriyanova, and F. Brännström, "Repair scheduling in wireless distributed storage with D2D communication," in *Proc. IEEE Inf. Theory Workshop (ITW)*, Oct. 2015, pp. 69–73.

[16] M. Gerami, M. Xiao, and M. Skoglund, "Partial repair for wireless caching networks with broadcast channels," *IEEE Wireless Commun. Lett.*, vol. 4, no. 2, pp. 145–148, Apr. 2015.

[17] J. Yang and S. Ulukus, "Optimal packet scheduling in an energy harvesting communication system," *IEEE Trans. Commun.*, vol. 60, no. 1, pp. 220–230, Jan. 2012.

[18] K. Tutuncuoglu and A. Yener, "Optimum transmission policies for battery limited energy harvesting nodes," *IEEE Trans. Wireless Commun.*, vol. 11, no. 3, pp. 1180–1189, Mar. 2012.

[19] O. Ozel, K. Tutuncuoglu, J. Yang, S. Ulukus, and A. Yener, "Transmission with energy harvesting nodes in fading wireless channels: Optimal policies," *IEEE J. Sel. Areas Commun.*, vol. 29, no. 8, pp. 1732–1743, Sep. 2011.

[20] J. Yang, O. Ozel, and S. Ulukus, "Broadcasting with an energy harvesting rechargeable transmitter," *IEEE Trans. Wireless Commun.*, vol. 11, no. 2, pp. 571–583, Feb. 2012.

[21] J. Yang and S. Ulukus, "Optimal packet scheduling in a multiple access channel with energy harvesting transmitters," *J. Commun. Netw.*, vol. 14, no. 2, pp. 140–150, Apr. 2012.

[22] K. Tutuncuoglu and A. Yener, "Sum-rate optimal power policies for energy harvesting transmitters in an interference channel," *J. Commun. Netw.*, vol. 14, no. 2, pp. 151–161, Apr. 2012.

[23] D. Gunduz and B. Devillers, "Two-hop communication with energy harvesting," in *Proc. IEEE Int. Workshop Comput. Adv. Multi-Sensor Adapt. Process. (CAMSAP)*, Dec. 2011, pp. 201–204.

[24] K. Tutuncuoglu, B. Varan, and A. Yener, "Throughput maximization for two-way relay channels with energy harvesting nodes: The impact of relaying strategies," *IEEE Trans. Commun.*, vol. 63, no. 6, pp. 2081–2093, Jun. 2015.

[25] B. Devillers and D. Gunduz, "A general framework for the optimization of energy harvesting communication systems with battery imperfections," *J. Commun. Netw.*, vol. 14, no. 2, pp. 130–139, Apr. 2012.

[26] O. Orhan, D. Gunduz, and E. Erkip, "Energy harvesting broadband communication systems with processing energy cost," *IEEE Trans. Wireless Commun.*, vol. 13, no. 11, pp. 6095–6107, Nov. 2014.

[27] K. Tutuncuoglu, A. Yener, and S. Ulukus, "Optimum policies for an energy harvesting transmitter under energy storage losses," *IEEE J. Sel. Areas Commun.*, vol. 33, no. 3, pp. 467–481, Mar. 2015.

[28] Y. Dong, F. Farnia, and A. Ozgur, "Near optimal energy control and approximate capacity of energy harvesting communication," *IEEE J. Sel. Areas Commun.*, vol. 33, no. 3, pp. 540–557, Mar. 2015.

[29] P. Blasco and D. Gunduz, "Multi-access communications with energy harvesting: A multi-armed bandit model and the optimality of the myopic policy," *IEEE J. Sel. Areas Commun.*, vol. 33, no. 3, pp. 585–597, Mar. 2015.

[30] S. Ulukus, A. Yener, E. Erkip, O. Simeone, M. Zorzi, P. Grover, and K. Huang, "Energy harvesting wireless communications: A review of recent advances," *IEEE J. Sel. Areas Commun.*, vol. 33, no. 3, pp. 360–381, Mar. 2015.

[31] C.-A. Chen, M. Won, R. Stoleru, and G. G. Xie, "Energy-efficient fault-tolerant data storage and processing in mobile cloud," *IEEE Trans. Cloud Comput.*, vol. 3, no. 1, pp. 28–41, Mar. 2015.

[32] Y. Zhang and N. Ansari, "Green data centers," *Handbook Green Inf. Commun. Syst.*, pp. 331–352, Nov. 2012.

[33] C. M. Vigorito, D. Ganesan, and A. G. Barto, "Adaptive control of duty cycling in energy-harvesting wireless sensor networks," in *Proc. IEEE 4th Annu. Commun. Soc. Conf. Sensor Mesh Ad-Hoc Commun. Netw. (SECON)*, Jun. 2007, pp. 21–30.

[34] X. Fafoutis, D. Vuckovic, A. Di Mauro, N. Dragoni, and J. Madsen, "Energy-harvesting wireless sensor networks," in *Proc. 9th Europ. Conf. Wireless Sensor Netw. (EWSN)*, Feb. 2012, pp. 84–85.

[35] A. G. Dimakis, V. Prabhakaran, and K. Ramchandran, "Ubiquitous access to distributed data in large-scale sensor networks through decentralized erasure codes," in *Proc. IEEE 4th Int. Sym. Inf. process. Sensor Netw. (IPSN)*, Apr. 2005, pp. 111–117.

[36] A. G. Dimakis and K. Ramchandran, "Network coding for distributed storage in wireless networks," in *Networked Sensing Information and Control*. Springer, 2008, pp. 115–134.

[37] R. Impagliazzo, S. Lovett, R. Paturi, and S. Schneider, "0-1 integer linear programming with a linear number of constraints," in *Proc. Electron. Colloq. Comput. Complex. (ECCC)*, vol. 24, Feb. 2014, pp. 1–5.

[38] S. Bradley, A. Hax, and T. Magnanti, *Applied mathematical programming*. Addison Wesley, USA, Jan. 1977.

[39] J. B. Orlin, A. P. Punnen, and A. S. Schulz, "Integer programming: optimization and evaluation are equivalent," in *Algorithms and Data Structures*. Springer, Aug. 2009, pp. 519–529.

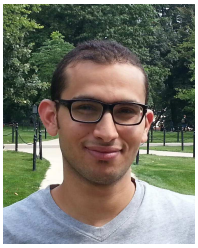
[40] D. Bertsekas, *Dynamic programming and optimal control*. Athena Scientific Belmont, MA, 2005, vol. 1, no. 3.

[41] V. T. Van, C. Yuen, and J. Li, "Non-homogeneous distributed storage systems," in *Proc. IEEE 50th Annu. Allerton Conf. Commun. Control Comput.*, Oct. 2012, pp. 1133–1140.

[42] J. Pernas, C. Yuen, B. Gastón, and J. Pujol, "Non-homogeneous two-rack model for distributed storage systems," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jul. 2013, pp. 1237–1241.

[43] D. S. Papailiopoulos and A. G. Dimakis, "Locally repairable codes," *IEEE Trans. Info. Theory*, vol. 60, no. 10, pp. 5843–5855, Oct. 2014.

[44] W. Song, S. H. Dau, C. Yuen, and T. J. Li, "Optimal locally repairable linear codes," *IEEE J. Sel. Areas Commun.*, vol. 32, no. 5, pp. 1019–1036, Oct. 2014.



Abdelrahman M. Ibrahim (S'07) received the B.Sc. degree in electrical engineering from Alexandria University, Alexandria, Egypt, in 2011, and the M.Sc. degree in wireless communications from Nile University, Giza, Egypt, in 2014. He is currently pursuing the Ph.D. degree and has been a graduate research assistant with the Department of Electrical Engineering, Pennsylvania State University, University Park, PA, since 2014. His research interests include green data storage systems, energy harvesting, and resource allocation for wireless networks.

Ahmed A. Zewail (S'07) received the B.Sc. degree in electrical engineering from Alexandria University, Alexandria, Egypt, in 2011, and the M.Sc. degree in wireless communications from Nile University, Giza, Egypt, in 2013. He is currently pursuing the Ph.D. degree with the Wireless Communications and Networking Laboratory (WCAN), Pennsylvania State University, University Park, PA. His research interests include wireless communication, capacity of relay networks, and information theoretic security.

Aylin Yener (S'91–M'00–SM'13–F'14) received the B.Sc. degree in electrical and electronics engineering and the B.Sc. degree in physics from Bogazici University, Istanbul, Turkey, and the M.S. and Ph.D. degrees in electrical and computer engineering from Wireless Information Network Laboratory (WINLAB), Rutgers University, New Brunswick, NJ, USA. She is a Professor of Electrical Engineering at The Pennsylvania State University, University Park, PA, USA, since 2010, where she joined the Faculty as an Assistant Professor in 2002. During the academic year 2008 to 2009, she was a Visiting Associate Professor with the Department of Electrical Engineering, Stanford University, Stanford, CA, USA. Her research interests include information theory, communication theory, and network science, with recent emphasis on green communications and information security. She received the NSF CAREER award in 2003, the Best Paper Award in Communication Theory in the IEEE International Conference on Communications in 2010, the Penn State Engineering Alumni Society (PSEAS) Outstanding Research Award in 2010, the IEEE Marconi Prize Paper Award in 2014, the PSEAS Premier Research Award in 2014, and the Leonard A. Doggett Award for Outstanding Writing in Electrical Engineering at Penn State in 2014.

Dr. Yener is currently a member of the Board of Governors of the IEEE Information Theory Society, where she was previously the treasurer (2012–2014). She served as the Student Committee Chair for the IEEE Information Theory Society 2007–2011, and was the Co-founder of the Annual School of Information Theory in North America co-organizing the school in 2008, 2009, and 2010. She was a Technical (Co)-Chair for various symposia/tracks at the IEEE ICC, PIMRC, VTC, WCNC, and Asilomar (2005–2014). She served as an Editor for the IEEE TRANSACTIONS ON COMMUNICATIONS (2009–2012), an Editor and an Editorial Advisory Board Member for the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS (2001–2012), and a Guest Editor for the IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY (2011) and the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS (2015).