# Repair Strategies for Mobile Storage Systems

Gokhan Calis, *Member, IEEE,* Swetha Shivaramaiah, O. Ozan Koyluoglu, *Senior Member, IEEE,* and Loukas Lazos, *Member, IEEE*

**Abstract**—We study the data reliability problem for devices forming a dynamic distributed storage system. Such systems are commonplace in traditional cloud storage applications where storage node failures and updates are frequent. We consider the application of regenerating codes for file maintenance. Such codes require lower bandwidth to regenerate lost data fragments compared to file replication or reconstruction. We investigate threshold-based repair strategies where data repair is initiated after a threshold number of data fragments have been lost. We show that at a low departure-to-repair rate regime, in which repairs are initiated after several nodes have left the system outperforms if repairs are initiated after a single node departure. This optimality is reversed when the node turnover is high. We further compare distributed and centralized repair strategies and derive the optimal repair threshold for minimizing the average repair cost per unit of time. In addition, we examine cooperative repair strategies and show performance improvements. We investigate several models for the time needed for node repair including a simple fixed time model and a more realistic model that takes into account the number of repaired nodes. Finally, an extended model where additional failures are allowed during the repair process is investigated.

**Index Terms**—Distributed storage, regenerating codes, dynamic cloud, mobile cloud, data reliability.

## 1 Introduction

Distributed storage systems (DSS) offer a high degree of reliability by replicating or coding data across multiple storage nodes [1], [2]. If some limited number of storage nodes fail, the original content can be recovered by downloading data fragments from healthy storage nodes. The amount of data that needs to be downloaded for repair is typically referred to as the *repair bandwidth*. In almost all DSS scenarios, the set of devices that comprise the system is dynamic. In traditional cloud storage applications, the number of storage nodes varies over time, as storage nodes are frequently taken offline due to failure or to apply updates [3]. The repair bandwidth is a substantial fraction of the overall network traffic generated within datacenters and a major driver of datacenter costs [4]. In other emerging DSS paradigms such as edge computing [5] and mobile cloud storage systems [6]–[8], the storage node dynamics can be higher as devices frequently enter and leave the system.

Reliability is one of the challenges in these networks [9]. Also, caching at the edge has been proposed as an effective approach to reduce backhaul usage and latency in content retrieval and gains using coding techniques are significant in such scenarios [10]. The so-called mobile cloud storage systems reduce the traffic load of the already over-burdened infrastructure network and improve content availability in the event of network outages. Furthermore, local caching and content distribution from a community of mobile devices may be employed, when backhaul connectivity is intermittent [6]–[8]. As an example, consider a tactical network such as a temporary military camp or tactical teams being deployed in the field. Soldiers carrying mobile devices may store information in a distributed fashion to improve reliability and reduce the risk of information exposure in the event of a capture. Another relevant application domain is that of delay tolerant networks (DTNs). Many developing regions in the world rely on networks where the reachback to infrastructure is intermittent and with high delay (could be in the order of days). The majority of users in those areas have access to mobile devices rather than a stationary storage device such as a desktop computer. Storing data in a mobile DSS can i increase the storage reliability while providing efficient access.

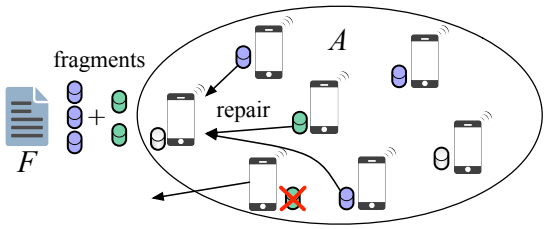



Fig. 1: File maintenance in a mobile cloud storage system.

In this paper, we consider a dynamic DSS setup where nodes leave and enter the DSS leading to partial data loss. We focus on a mobile cloud scenario where data is stored within a geographically-limited area $\mathcal{A}$ by a community of mobile devices, as shown in Fig. 1. For ease of illustration, we consider a single file $\mathcal{F}$. A user within $\mathcal{A}$ can download $\mathcal{F}$ from the mobile devices via direct communication links without accessing the network infrastructure. When a mobile device storing any fragment of $\mathcal{F}$ exits $\mathcal{A}$, its stored data is lost. To deal with such losses, redundancy is introduced in the form of data replication or coding [1], [2]. In replication storage, copies of $\mathcal{F}$ are stored at multiple devices within the community. More sophisticated coding schemes such as erasure coding achieve the same reliability at lower storage overhead [12], [13]. Despite the application of coding, a stored file $\mathcal{F}$ will eventually be lost when enough mobile devices (storage nodes) depart from $\mathcal{A}$. To maintain $\mathcal{F}$ over long time periods, the mobile cloud system must be capable of recovering the lost data. A repair scenario is shown in Fig. 1. Lost data is recovered by downloading fragments from the storage nodes that remain within $\mathcal{A}$.

When multiple files are stored within $A$, the code used to reliably store files can differ depending on the file type and

size. In fact, it is common for DSS to use different encodings with different access frequency (hot vs. cold data), which an be dynamically updated [11]. Our framework expresses the optimal file maintenance strategy as a function of the code parameters and can therefore be applied to any regeneration code. The repair strategy can be individualized for every file type based on the applied code and the storage node dynamics. In the multiple file case, additional signaling overhead (e.g., a beaconing protocol) is required to maintain the state of each file, either at a leader node, or distributed.

The file maintenance problem for distributed storage systems has been primarily studied assuming that erasure codes are applied for redundancy [12], [14]. However, erasure codes are not repair bandwidth-efficient. The repair bandwidth can be reduced by applying regenerating codes, which allow fragment recovery without file reconstruction (see [15]–[18] and references therein). Although regenerating codes lower the repair bandwidth (per single node repair), the design of an efficient repair strategy for a dynamic DSS involves cost optimizations with respect to many parameters, including the code redundancy factor, the node departure and fragment repair rates, the threshold for initiating repair operations, and the available network bandwidth. In this paper, *we study the problem of minimizing the file maintenance cost, as a function of the network dynamics, the code parameters, and the communication model for repairing lost data fragments*. Specifically, we make the following contributions.

- We focus on threshold-based file maintenance strategies, in which repairs are initiated when a threshold number of fragments is lost. We analyze two communication models, namely *distributed repair* and *centralized repair*. In distributed repair, the new storage nodes independently download data from existing nodes to recover lost fragments. In centralized repair, a *leader* node first recovers $\mathcal{F}$ via reconstruction, before regenerating and distributing the repaired fragments to new storage nodes. In both scenarios, we assume that repairs are performed in parallel and there are no additional failures during fragment recovery. This simplified model allows us to derive closed-form expressions.
- We derive the *optimal repair threshold* that minimizes the average repair cost per unit of time for each communication model. Our results show that no one strategy is optimal for all possible system configurations and mobility patterns. At the low departure-to-repair rate regime, repairing at the regeneration threshold yields the optimal strategy. On the other hand, at the high departure-to-repair rate regime, regenerating after a single fragment loss minimizes the average repair cost per unit of time.
- We further investigate the application of cooperative repair codes. We show that the repair bandwidth is minimized at full cooperation, i.e., when all nodes to be repaired cooperate. We then investigate the centralized repair of multiple node failures, which suits our centralized repair model that is described earlier. The advantage of such a model is that a leader node does not need to download the file $\mathcal{F}$, which reduces the average repair cost per time.
- We revise the fixed-rate repair model originally assumed in distributed and centralized repair with a more realistic node-dependent model. In the latter, the repair time depends on the number of nodes that are repaired. We compare the resulting average repair cost with our earlier model and show that in the low departure-to-repair rate regime, the simplified repair-all-at-one model faithfully approximates the node-dependent one.
- We further consider a distributed repair model that may involve additional failures during recovery. We express the average repair cost through a system of equations and verify our analytical findings through simulations. Lastly, we compare all of the discussed distributed repair models employing regenerating codes.
- Our results indicate that the optimal threshold depends on the departure-to-repair rate ratio and the underlying code parameters. Furthermore, the average repair cost depends on the departure-to-repair rate ratio as well as the underlying code parameters in all scenarios discussed in the paper.

We emphasize that the applicability of our framework extends beyond mobile storage system to any DSS where fragment losses can occur. This includes popular wired distributed storage architectures such as HDFS [3]. For the wired domain, the departure-to-repair ratio represents the dynamics between fragment loss due to storage node failure, misconfiguration, or update and the rate of repair. A notable difference between a wired DSS and the mobile scenario is the expected regime for the departure-to-repair ratio. The mean-time-to-failure (MTTF) for the wired DSS is in the order of months [19] whereas a mobile DSS can experience failures (departures) at a much higher rate. Nevertheless, our framework characterizes the optimal repair strategy for any rate regime.

The remainder of the paper is organized as follows. In Section 2, we present related work. The system model is presented in Section 3. We analyze threshold-based file maintenance strategies in Section 4 and analytically compare them in Section 5. In Section 6, we analyze codes with cooperative repair and in Section 7, we extend the repair process model to one that considers the number of nodes that are repaired. Node departures during the repair phase are considered in Section 8. We conclude in Section 9.

## 2 Related Work

### 2.1 Coding in Distributed Storage

In reliable storage systems, information is replicated or coded such that the original content can be recovered if some limited fraction of the stored data is lost. Replication is the most intuitive way to introduce redundancy. This method refers to the maintenance of verbatim copies of the same file $\mathcal{F}$. Although replication is easy to implement, it suffers from high storage and repair overhead.

#### 2.1.1 Erasure Codes

Erasure codes incur less storage overhead compared to replication while maintaining the same degree of reliability.

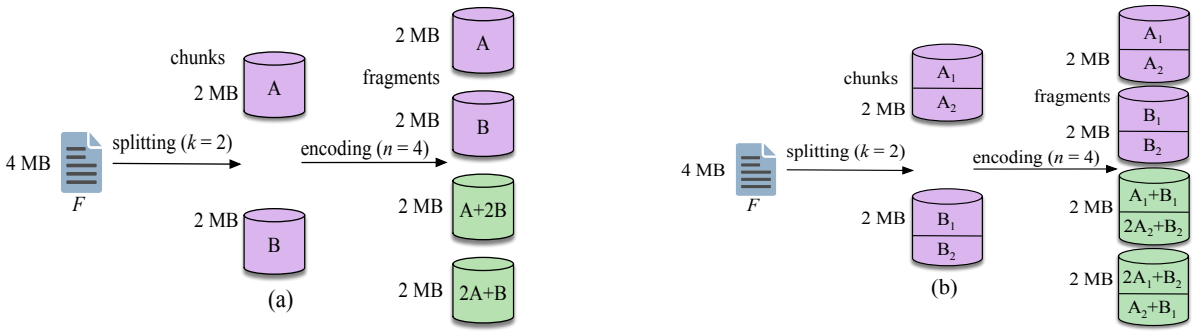



Fig. 2: Storage of $F$ using (a) a $(n = 4, k = 2)$ erasure code and, (b) a $(n = 4, k = 2, d = 3, \alpha = 2, \beta = 1)$ regenerating code.

In particular, Maximum Distance Separable (MDS) codes achieve the optimal tradeoff between failure tolerance and storage overhead [20], [21]. An $(n, k)$ MDS code encodes $k$ data chunks to $n$ fragments and can tolerate up to $n - k$ fragment losses. Any $k$ encoded fragments can be used to reconstruct $\mathcal{F}$. Fig. 2(a) shows the encoding process for a file $\mathcal{F}$ of size 4MB using a (4, 2) erasure code. File $\mathcal{F}$ is split into $k = 2$ chunks $A$ and $B$, each of size 2MB. The two chunks are then encoded into $n = 4$ fragments. The repair bandwidth for this scheme equals the size of the original file. Reed-Solomon codes are a classical example of MDS codes and are deployed in many existing storage systems (e.g. [13], [22]–[24]).

#### 2.1.2 Regenerating Codes

Although erasure codes offer significant savings in storage, their repair bandwidth is suboptimal, a data amount equal to the file size must be retrieved to repair a single fragment. Regenerating codes, on the other hand, can recover lost fragments without reconstructing the entire file, at the expense of a small storage overhead. They were initially investigated in the seminal work of Dimakis *et al.* [15], which focuses on the following setup. A file $\mathcal{F}$ of size $\mathcal{M}$ symbols is encoded into $n$ fragments, each of size $\alpha$ symbols, such that (i) the file can be reconstructed from any $k$ fragments, and (ii) a lost fragment can be repaired by downloading $\beta \leq \alpha$ symbols from any $d \geq k$ fragments, resulting in a repair bandwidth of $\gamma = d\beta$. Dimakis *et al.* characterized the tradeoff between the per node storage ($\alpha$) and the repair bandwidth ($\gamma$) [15].

Fig. 2(b) shows an example of a $(n, k, d, \alpha, \beta) = (4, 2, 3, 2, 1)$ regenerating code. Here, the file $\mathcal{F}$ is split into $k = 2$ chunks each of size $\alpha = 2$MB. The chunks are encoded in $n = 4$ fragments, with each fragment being 2MB. A failed node in this scenario can be regenerated by retrieving fragments of size $\beta = 1$MB from $d = 3$ surviving nodes. This yields a repair bandwidth of $d\beta = 3$MB which is less that $k\alpha = 4$MB. Note, however, that regeneration can be applied only if at least $d$ fragments are available. If fewer than $d$ but more than $k$ fragments remain available, the lost fragments can only be repaired through file reconstruction.

During the repair process of regenerating codes, there is no coordination among the nodes to be repaired. In [12], [25]–[27], the authors consider the case where $t$ storage nodes are repaired simultaneously in a cooperative manner. Specifically, referring to this set of $t$ nodes as the newcomers, and the existing nodes storing fragments of $\mathcal{F}$ as live nodes, each newcomer contacts $d$ live nodes and downloads $\beta$ symbols from each. Moreover, newcomers cooperate and download $\beta'$ symbols from each of the remaining $t - 1$ newcomers. The tradeoff between per node storage and repair bandwidth is established similarly to [15]. Rawat *et al.* in [28] also consider the cooperative repair of $t$ nodes such that only one node among $t$ nodes downloads data from live nodes. After downloading the necessary information at the leader node, the remaining $t - 1$ nodes are cooperatively repaired. Two points corresponding to minimum storage and minimum bandwidth regeneration are characterized.

### 2.2 Maintaining Distributed Storage Systems

In the work of Dimakis *et al.* [15] and the following works for regenerating codes, the repair process usually refers to regenerating a single failed node, i.e., due to hardware failures. In the concept of mobile storage systems, a failed node not necessarily occur due to a hardware failure but it may also refer to a departure of the node from the mobile network. Nevertheless, in order to maintain the capability of such network, the lost information needs to be regenerated. Furthermore, in the literature of regenerating codes, there is often no preference to when to perform repair as eager repair is the de facto approach to any failure. In our work, we also consider the mobility rate of the storage systems and propose optimal thresholds at which the repair process is more efficient in terms of average cost of data transfer over network per time.

In [29], [30], chubby local services for Google File System and ZooKeeper for Yahoo! introduce coordination schemes to handle large-scale systems. Our work here can be considered as a complimentary study on how to maintain data availability in the case of mobility and therefore can be combined with such schemes. However, it should be noted that such systems may require a *master* node/server, which resembles to our study on centralized repair.

In the context of mobile cloud systems, Pääkkönen *et al.* considered a wireless device-to-device network used for distributed storage [31]. The authors showed the energy consumption for maintaining data using regenerating codes is lower compared to retrieving a lost file from a remote source. This result holds if the per-bit energy cost for communication between the mobile devices is lower than the cost for communicating with the remote source.

In a follow-up work, Pääkkönen *et al.* compared replication with regeneration for a similar wireless P2P storage system [7]. They derived closed-form expressions for the expected total energy cost of file retrieval using replication and regeneration. They showed that the expected total cost of 2-

replication is lower than the cost of regeneration. However, only an eager repair strategy was considered in the analysis. Moreover, the advantages of regeneration were not fully exploited by considering codes with different parameters. Pääkkönen *et al.* also addressed the problem of tolerating multiple simultaneous failures [32]. They investigated the energy overhead of regenerating codes in a cellular network. They showed that large energy gains can be obtained by employing regenerating codes. These gains depend on the file popularity. The authors provided decision rules for choosing between simple caching, replication, regenerating codes, based on numerical results on certain application scenarios. In our work, we analytically provide decision rules to choose optimal repair strategies that minimize the repair bandwidth per unit of time. One additional benefit of minimizing the repair bandwidth is that fewer repairs are required, which also lowers the encoding/decoding needs that can be computationally intensive. Although we do not precisely characterize the impact of such costs, the optimal repair strategies we propose implicitly lowers them.

Pedersen *et al.* recently studied the cost of content caching on mobile devices using erasure codes [33]. They derived analytical expressions for the cost of content download and repair bandwidth as a function of the repair interval. These expressions were used to evaluate the communication cost of distributed storage for MDS codes, regenerating codes, and locally repairable codes. Their results show that in high churn, distributed storage can reduce the communication cost compared to downloading from a base station. They conclude that MDS codes are the best performers in this setup.

## 3 System Model

### 3.1 Network Model

We consider a distributed storage system (DSS) consisting of mobile storage nodes that enter and exit a geographically-limited area $\mathcal{A}$. When a node departs from $\mathcal{A}$, its data is lost. The nodes that store file fragments within area $\mathcal{A}$ are said to be *live nodes*. New nodes that are used to store repaired fragments are said to be *newcomer nodes* or newcomers. We assume that there are always sufficient newcomers to perform repairs. Moreover, as we are interested in the system performance due to network dynamics, we do not consider data loss due to hardware failures. Such failures occurs orders of magnitude less frequently than node departures (in the order of 4.3 months for wired DSS [19]). Following the network dynamics model of prior works [14], [31], we model the time $X_i$ spent by each node within $\mathcal{A}$ as an exponentially distributed random variable with parameter $\lambda$ (i.e., $X_i \sim \text{Exp}(\lambda),\ \forall i$). Random variables $\{X_i\}$ are assumed independent and identically distributed.

The repair time is modeled by an exponentially distributed random variable with parameter $\mu$. For ease of analysis, we initially assume that $\mu$ is independent of the number of fragments that need to be repaired. We later revise our analysis and consider a more realistic model in which repairs proceed in parallel at different nodes with the same rate $\mu$. This corresponds to the distributed nature of mobile DSS. Finally, we define $\rho = \frac{\lambda}{\mu}$ as the ratio of the departure-to-repair rate.

### 3.2 Storage Model

A file $\mathcal{F}$ of size $\mathcal{M}$ bits is stored in $n$ storage nodes using a regenerating code with parameters $(n, k, d, \alpha, \beta)$ (see Fig. 2(b)). We focus on the two most popular types of regenerating codes, namely Minimum Storage Regenerating (MSR) codes and Minimum Bandwidth Regenerating (MBR) codes. These two classes of codes operate at the end points of the tradeoff between per node storage and repair bandwidth, as introduced in [15]. MSR codes achieve minimum storage by setting $\alpha = \mathcal{M}/k$ and minimize the repair bandwidth under this constraint. Their operating point is:

$$(\alpha_{\text{MSR}}, \gamma_{\text{MSR}}) = \left(\frac{\mathcal{M}}{k}, \frac{\mathcal{M}d}{k\,(d-k+1)}\right). \quad (1)$$

Note that, for MSR codes, $\alpha_{\text{MSR}} \leq \gamma_{\text{MSR}}$ and hence, the per-node storage is smaller than the repair bandwidth. MBR codes, on the other hand, minimize the repair bandwidth (achieved when $\gamma = \alpha$), and operate at:

$$(\alpha_{\text{MBR}}, \gamma_{\text{MBR}}) = \left(\frac{2\mathcal{M}d}{2kd-k^2+k}, \frac{2\mathcal{M}d}{2kd-k^2+k}\right). \quad (2)$$

Instances of these codes can be found in [16]–[18]. We note that although we treat a single file for ease of illustration, the same methodology can be extended to multiple files by scaling the corresponding operating points accordingly. Furthermore, although we focus on only MSR and MBR points of the tradeoff curve, our results can be replicated for any other operating point on the curve.

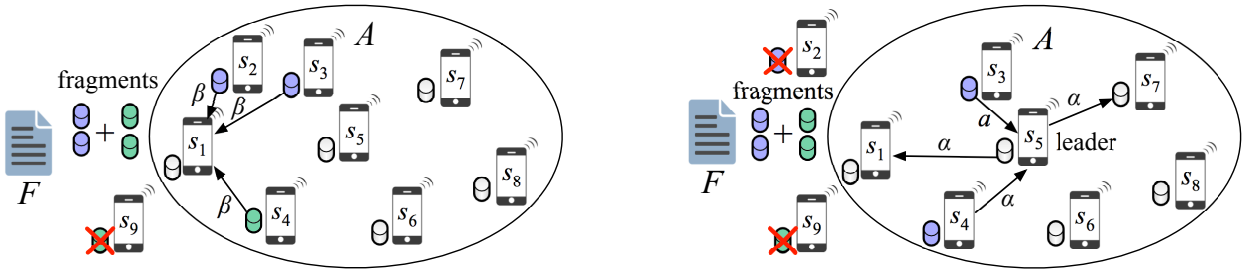
### 3.3 File Repair Model

In our model, the system continuously monitors the redundancy level and initiates a repair when $\tau$ live nodes remain within $\mathcal{A}$. The determination of $\tau$, the type of repair (regeneration, reconstruction, or both) and the communication model for fragment retrieval (centralized or distributed) form a *file maintenance strategy*. We note that the practical implementation details of the redundancy monitoring mechanism and of the communication protocols for retrieving various fragments are beyond the scope of the present work. For example, there may be an additional communication overhead depending on the file repair model. We focus on the theoretical aspects of the maintenance process. Since repairs are initiated only when the number of remaining nodes reaches threshold $\tau$, a repair strategy can be viewed as an i.i.d. system recovery process occurring every $\Delta$ seconds, where $\Delta$ is a random variable denoting the time elapsed between two instances of a fully repaired system. For this recovery process, we define the following costs.

**Definition 1** (Repair cost $c(\tau)$)**.** *The number of bits $c(\tau)$ that must be downloaded from the $\tau$ remaining nodes to restore $n$ fragments in $\mathcal{A}$, when $n-\tau$ nodes have departed $\mathcal{A}$.*

**Definition 2** (Average repair cost per unit of time $r(\tau)$)**.** *The average cost per unit of time for maintaining $n$ fragments in $\mathcal{A}$, defined as $c(\tau)$ over the average time between two instances of a fully repaired system, i.e., $E[\Delta]$, with $n$ fragments ($r(\tau)$ is measured in bits per unit of time).*

The distributed storage systems are susceptible to failures and in order to maintain such systems, it's important




(a) distributed repair (b) centralized repair

Fig. 3: (a) Distributed repair: nodes independently regenerate a lost fragment by obtaining symbols from other nodes, (b) centralized repair: a leader node reconstructs $\mathcal{F}$ and distributes lost fragments to new nodes.

to regenerate the lost information due to the departed nodes by transferring data to a newcomer node. As a result, we want to minimize the required data transfer per time which is defined above. We determine the optimal file maintenance strategy for different node departure rates, code parameters, and communication models for fragment retrieval.

Our analysis focuses on the repair bandwidth due to the mobility dynamics and does not consider the storage-repair bandwidth tradeoff, as addressed in prior works [15]. The storage overhead is fixed by the code selection. We aim at optimizing the repair overhead, once the code is determined, and also study the impact of node mobility on the repair process.

## 4 File Maintenance Strategies

Let $\tau$ denote the number of live nodes remaining within $\mathcal{A}$ after the departure of $n-\tau$ nodes. We focus on determining the optimal repair threshold $\tau^*$, which minimizes the average repair cost per unit of time. We first compare the distributed repair strategy with centralized repair strategy.

### 4.1 Distributed Repair

In distributed repair, newcomers recover lost fragments by independently downloading relevant symbols from live nodes. The repair process is initiated when $\tau$ live nodes remain within $\mathcal{A}$, where $k \leq \tau < n-1$ (when $\tau < k$, the data is irrecoverably lost). If $\tau \geq d$, fragment recovery can be performed through regeneration. Each of the $n-\tau$ newcomers downloads $\beta$ symbols from $d$ live nodes and independently regenerates a lost fragment. Fig. 3(a) demonstrates the distributed repair process for a file $\mathcal{F}$ stored with a $(n=4, k=2, d=3, \alpha=2, \beta=1)$ regenerating code. One fragment of $\mathcal{F}$ is lost because node $s_9$ departed from $\mathcal{A}$. The lost fragment is regenerated at $s_1$ by independently downloading $\beta=1$ symbol from three nodes. The total repair bandwidth is equal to 3 symbols.

If $\tau < d$, regeneration cannot be directly applied. To reduce the repair cost, we consider a hybrid scheme consisting of regeneration and reconstruction. First, $d-\tau$ nodes are repaired by downloading $\alpha$ symbols from $k$ live nodes and reconstructing $\mathcal{F}$. When $d$ fragments become available, regeneration is applied to repair the remaining $n-d$ newcomers. Accordingly, the repair cost is expressed by:

$$c_D(\tau) = \begin{cases} k\alpha(d-\tau) + \gamma(n-d), & \text{if } \tau < d \\ \gamma(n-\tau), & \text{if } \tau \geq d. \end{cases} \quad (3)$$

The subscript $D$ in $c_D(\tau)$ is used to denote the cost of distributed repair and $\gamma$ denotes the regeneration cost of a single fragment which depends on the underlying regeneration code (see eqs. (1) and (2) for MSR and MBR codes, respectively). From (3), it is evident that $c_D(\tau)$ monotonically decreases with $\tau$. Moreover, the rate of cost change (with respect to $\tau$) is higher when $\tau < d$. To determine the optimal threshold $\tau^*$, we minimize $r_D(\tau)$, which captures the repair cost for maintaining $n$ fragments per unit of time.

To calculate $r_D(\tau)$, we use the continuous-time Markov chain (CTMC) model shown in Fig. 4. This model captures the periodic repair process when node departures occur independently, the time spent by each node in $\mathcal{A}$ is exponentially distributed with parameter $\lambda$, and the system recovery process is exponentially distributed with parameter $\mu$.

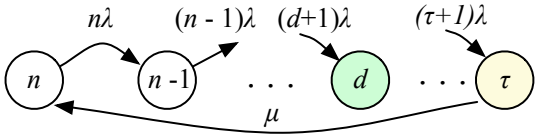



Fig. 4: Markov chain for a threshold-based file maintenance.

The CTMC consists of $n-\tau+1$ states representing the number of fragments that remain within $\mathcal{A}$ after each node departure, until a repair at state $\tau$ is initiated. Note that we have omitted states after $\tau$ in the CTMC model, because we are interested in optimizing the periodic cost of repairing the DSS at threshold $\tau$. Moreover, the transition probability to state $\tau-1$ is negligible for most realistic scenarios in which $\mu \gg \tau\lambda$. For cases when $\mu \not\gg \tau\lambda$, we compute the mean time it takes to depart from the optimal repair strategy of repairing at state $\tau$ and interpret this event as a form of system error which leads to data loss (see Section 5.3).

For the CTMC in Fig 4, the departure rate from a state $i$ equals the node departure rate $\lambda$, times the number of nodes which store fragments at state $i$. When the repair process is initiated, the system transitions from state $\tau$ to state $n$ because all fragment repairs nodes proceed in parallel. For the CTMC, we define the expected average cost $r_D(\tau)$ per unit of time as

$$r_D(\tau) = \frac{c_D(\tau)}{\mathrm{E}[\Delta]}, \quad (4)$$

where $E[\Delta]$ is the average time between two transitions

through the $n^{th}$ state in the periodic repair process[1]. For $\Delta$,

$$\Delta = T_n + T_{n-1} + \cdots + T_{\tau+1} + T_\tau, \tag{5}$$

where $T_i$ denotes the time that the system stays at state $i$ (inter-departure time) and $T_\tau$ is the expected time for completing repairs so that $n-\tau$ fragments are recovered (return to state $n$). The random variables $T_i$ are independent and exponentially distributed with parameter $i\lambda$, whereas $T_\tau$ is exponentially distributed with parameter $\mu$. In particular, $\mathrm{E}[T_i] = \frac{1}{i\lambda}$ and $\mathrm{E}[T_\tau] = \frac{1}{\mu}$. Therefore, $\mathrm{E}[\Delta]$ is the sum expectation of independent exponential random variables.

$$\mathrm{E}[\Delta] = \sum_{i=\tau+1}^{n} \frac{1}{i\lambda} + \frac{1}{\mu} = \frac{H_{n,\tau}}{\lambda} + \frac{1}{\mu}, \tag{6}$$

where $H_{n,\tau} = \sum_{i=\tau+1}^{n} \frac{1}{i}$. Combining (4) and (6), we obtain the average repair cost per unit of time as follows.

$$r_D(\tau) = \frac{c_D(\tau)}{\mathrm{E}[\Delta]} = \begin{cases} \frac{\lambda\mu(k\alpha(d-\tau)+\gamma(n-d))}{\mu H_{n,\tau}+\lambda}, & \text{if } \tau < d \\ \frac{\lambda\mu(\gamma(n-\tau))}{\mu H_{n,\tau}+\lambda}, & \text{if } \tau \geq d. \end{cases} \tag{7}$$

We use (7) to determine the optimal threshold $\tau^*$ which minimizes $r_D(\tau)$. This is given by Propositions 1 and 2.

**Proposition 1.** *For regeneration ($d \leq \tau \leq n-1$), the optimal repair threshold $\tau^*$ is given by*

$$\tau^* = \begin{cases} d, & \rho \leq \frac{H_{n-1,d}}{n-d-1} - \frac{1}{n} \\ n-1, & \textit{otherwise.} \end{cases} \tag{8}$$

*Proof.* Proof is provided in Appendix A. □

Proposition 1 determines the $\rho$ regime for which repairs at $\tau = d$, an instance of *lazy repair*, is more efficient than initiating repairs at $\tau = n-1$, referred to as *eager repair*. If the departure-to-repair ratio is low, that means we are not likely to see departures from the network often, therefore there is no urgency to repair a failed node. As a result, we can tolerate to wait for multiple nodes to depart before we initiate the repair process. On the other hand, if $\rho$ is high, that means that we cannot tolerate multiple nodes to be failed at a given time because high $\rho$ suggests it's likely to have more node departures before a repair process is finished. Therefore, an eager repair scheme is more suitable in such cases. In the following Lemma, we show that there is always a positive $\rho$ for which lazy repair is more efficient that eager repair.

**Lemma 1.** *There is always some $\rho > 0$ for which lazy repair ($\tau^* = d$) is more efficient than eager repair ($\tau = n-1$), independent of the code parameters used for regeneration.*

*Proof.* Proof is provided in Appendix B. □

We now examine if there is a $\rho$ regime for which the hybrid scheme, i.e., reconstruction plus regeneration results in a lower expected cost per unit of time compared to regeneration only. This rate regime is given by the following proposition.

1. The alternative definition of $r_D(\tau) = E\left[\frac{c_D(\tau)}{\Delta}\right]$ is not useful because the expectation is infinite. This is due to the infinitesimally small values that can be obtained by $\Delta$, whereas $c_D(\tau)$ remains lower bounded.

**Proposition 2.** *For regeneration plus reconstruction ($k \leq \tau \leq d$), the optimal repair threshold $\tau^*$ is given by*

$$\tau^* = \begin{cases} k, & \rho \leq \frac{\gamma(n-d)H_{d,k}}{k\alpha(d-k)} - H_{n,d} \\ d, & \textit{otherwise.} \end{cases} \tag{9}$$

*Proof.* Proof is provided in Appendix C. □

Similar to Lemma 1, we investigate if the highest departure-to-repair rate for which reconstruction at $k$ is more efficient than regeneration is always positive independent of the code parameters. Unlike the case of Lemma 1, we show that for a certain relationship between $n, k, \gamma$, and $\alpha$, regeneration is strictly more efficient than regeneration plus reconstruction, independent of $\rho$. For any other code parameters, the most efficient strategy depends on $\rho$.

**Lemma 2.** *For any departure-to-repair ratio $\rho$, regeneration is strictly more efficient than regeneration plus reconstruction for codes satisfying $n\gamma < k^2\alpha$.*

*Proof.* Proof is provided in Appendix D. □

We further explore the condition in Lemma 2 for MSR and MBR codes. For MSR codes, we obtain that $dn < k^2(d-k+1)$ by substituting the operation points of MSR from (1). Similarly, for MBR codes, we obtain that $n < k^2$ by substituting the operation points of MBR from (2). Note that Lemma 2 does not enumerate all possible codes for which regeneration is strictly more efficient than regeneration plus reconstruction for any $\lambda$. This is because we have used bounds on the harmonic function to derive the analytic formulas. Numerical bounds could provide a more accurate range of code parameters for which Lemma 2 is true.

**Metadata overhead:** In our analysis thus far, we have ignored the practical implementation details related to synchronizing repairs. These relate to maintaining the list of live nodes and newcomers and can be achieved by a periodic beaconing operation announcing the state of each node, which is already implicit in mobile networks. When the number of live nodes falls below the optimal threshold, $n-\tau^*$ newcomer nodes can request $d\beta$ symbols from *any* live nodes and restore $\mathcal{F}$ and update the list of live nodes. We note that no metadata is needed for which symbols are stored at each node. This is because any $d$ fragments can be used for reconstruction. Moreover, the code structure can be regenerated by known the code generator matrix. A similar metadata overhead is necessary for centralized repair.

### 4.2 Centralized Repair

In the centralized strategy, repairs are performed by a *leader node* in two stages. In the first stage, the leader downloads $\alpha$ symbols from $k$ live nodes and reconstructs $\mathcal{F}$. In the second stage, the leader node transmits $\alpha$ bits to each of the remaining $(n-\tau-1)$ newcomers to restore the remaining $(n-\tau-1)$ fragments. Similarly to distributed repair, the only metadata that is needed in centralized repair is maintaining a list of live nodes and newcomers at the leader node. Fig. 3(b) shows an example of centralized repair for a $(n=4, k=2, d=3, \alpha=2, \beta=1)$ regenerating code. Nodes $s_2$ and $s_9$ have departed from area $\mathcal{A}$, leading to the loss of their respective fragments. Node $s_5$, who acts as a

TABLE 1: Cost comparison of repair strategies at different thresholds.

| | Distributed Repair | | | Centralized Repair | |
|---|---|---|---|---|---|
| | Regeneration | | Regeneration + Reconstruction | Reconstruction | |
| Code | $r_D(n-1)$ | $r_D(d)$ | $r_D(k)$ | $r_C(n-1)$ | $r_C(k)$ |
| MSR | $\frac{n\mathcal{M}d\lambda\mu}{k(d-k+1)(\mu+n\lambda)}$ | $\frac{\mathcal{M}(n-d)d\lambda\mu}{k(d-k+1)(\lambda+\mu H_{n,d})}$ | $\frac{\mathcal{M}[k(d-k+1)(d-k)+d(n-d)]\lambda\mu}{k(d-k+1)(\lambda+\mu H_{n,k})}$ | $\frac{n\mathcal{M}\lambda\mu}{\mu+n\lambda}$ | $\frac{(n-1)\mathcal{M}\lambda\mu}{k(\lambda+\mu H_{n,k})}$ |
| MBR | $\frac{2n\mathcal{M}d\lambda\mu}{k(2d-k+1)(\mu+n\lambda)}$ | $\frac{2\mathcal{M}(n-d)d\lambda\mu}{k(2d-k+1)(\lambda+H_{n,d})}$ | $\frac{2\mathcal{M}d(n+kd-k^2-d)\lambda\mu}{k(2d-k+1)(\lambda+H_{n,k})}$ | $\frac{2n\mathcal{M}d\lambda\mu}{(2d-k+1)(\mu+n\lambda)}$ | $\frac{2(n-1)\mathcal{M}d\lambda\mu}{k(2d-k+1)(\lambda+\mu H_{n,k})}$ |

leader, downloads $\alpha = 2$ symbols from $k = 2$ other nodes to reconstruct $\mathcal{F}$. It then distributes $\alpha = 2$ symbols to $s_1$ and $s_7$ to restore the system reliability. The repair cost of centralized repair is given by:

$$c_C(\tau) = \alpha\,(k + n - \tau - 1)\,. \tag{10}$$

In (10), the subscript $C$ in $c_C(\tau)$ is used to denote the cost of centralized repair. The node departure process does not vary with the repair strategy. Therefore, the same CTMC model shown in Fig. 4 applies for the centralized repair. According to (4), the average repair cost $r_C(\tau)$ is given by:

$$r_C(\tau) = \frac{c_C(\tau)}{\mathrm{E}[\Delta]} = \frac{\lambda\mu\alpha(k + n - \tau - 1)}{\mu H_{n,\tau} + \lambda}. \tag{11}$$

The optimal threshold $\tau^*$ which minimizes $r(\tau)$ is obtained in Proposition 3.

**Proposition 3.** *The optimal repair threshold $\tau^*$ which minimizes $r(\tau)$ for centralized repair is given by*

$$\tau^* = \begin{cases} k, & \rho \leq \frac{kH_{n-1,k}}{n-k-1} - \frac{1}{n} \\ n-1, & \textit{otherwise} \end{cases} \tag{12}$$

*Proof.* Proof is provided in Appendix E. □

Using Proposition 3, we can determine the optimal repair strategy for any $\rho$, when centralized repair is employed. We note that according to Lemma 1, the value $\frac{kH_{n-1,k}}{(n-k-1)} - \frac{1}{n}$ is strictly positive for any code parameters. Therefore, there is always a departure-to-repair ratio for which lazy repair is more efficient than eager repair, independent of the code used for regeneration and reconstruction.

**Maintaining multiple files:** Let us consider the scenario where a total of $\Gamma$ files are stored in the DSS. In a homogeneous storage system, each file could be stored using the same regenerating code $\mathcal{C}$ (to accommodate different file sizes, files larger than the code length can be partitioned to several subfiles). When the number of live nodes falls below $\tau^*$, the repair process is initiated synchronously for all files, as the repair threshold is reached simultaneously for all files (live nodes store an equal number of fragments). In this case, the repair bandwidth is simply scaled by the number of files.

Now consider a heterogeneous scenario where a different code is applied to different file types (hot vs. cold files). This results in different repair thresholds $\tau_i^*$ for each file. In this case, each file can be repaired independently when the number of live nodes storing it falls below $\tau_i^*$. To facilitate independent repairs, additional metadata information needs to be stored to reflect *the number of live nodes per file*. The latter can be by updated for all files every time a node leaves the area $\mathcal{A}$, or a file is repaired. For centralized repair, this information is available at the leader node.

## 5 Analysis of Maintenance Strategies

In this section, we characterize the $\rho$ regime for which lazy repair is more cost-efficient than eager repair. Moreover, we determine the optimal repair strategy (decentralized vs. centralized) as a function of the code parameters, when the departure and repair rates are fixed. To ease the reader to our analysis, we summarize the cost of repair in Table 1.

### 5.1 Eager vs. Lazy Repair

According to the results of Propositions 1, 2, and 3, we classify the departure-to-repair ratios into a *low departure-to-repair rate regime* ($\rho_{low}$) and a *high departure-to-repair rate regime* ($\rho_{high}$). The two regimes are defined by finding the lowest and highest rates, based on the bounds stated in the three propositions.

$$\rho_{\text{low}} = \min\left\{\tfrac{H_{n-1,d}}{n-d-1} - \tfrac{1}{n}, \tfrac{\gamma(n-d)H_{d,k}}{k\alpha(d-k)} - H_{n,d}, \tfrac{kH_{n-1,k}}{(n-k-1)} - \tfrac{1}{n}\right\}. \tag{13}$$

$$\rho_{\text{high}} = \max\left\{\tfrac{H_{n-1,d}}{n-d-1} - \tfrac{1}{n}, \tfrac{\gamma(n-d)H_{d,k}}{k\alpha(d-k)} - H_{n,d}, \tfrac{kH_{n-1,k}}{(n-k-1)} - \tfrac{1}{n}\right\}. \tag{14}$$

Noting that $\frac{H_{n-1,d}}{n-d-1} - \frac{1}{n} < \frac{kH_{n-1,k}}{(n-k-1)} - \frac{1}{n}$ for $k < d$, the two regime expressions can be simplified to

$$\rho_{\text{low}} = \min\left\{\tfrac{H_{n-1,d}}{n-d-1} - \tfrac{1}{n}, \tfrac{\gamma(n-d)H_{d,k}}{k\alpha(d-k)} - H_{n,d}\right\}. \tag{15}$$

$$\rho_{\text{high}} = \max\left\{\tfrac{\gamma(n-d)H_{d,k}}{k\alpha(d-k)} - H_{n,d}, \tfrac{kH_{n-1,k}}{(n-k-1)} - \tfrac{1}{n}\right\}. \tag{16}$$

For any $\rho \leq \rho_{\text{low}}$, the repair cost per unit of time is minimized when lazy repair is applied since that choice of $\rho$ would be lower than the bounds found in (8), (9) and (12) and the corresponding repair thresholds are the lowest possible. On the other hand, for any $\rho \geq \rho_{\text{high}}$, eager repair (i.e., repair at $\tau^* = n - 1$) yields the lowest $r(\tau)$. These findings hold for both distributed and centralized repair. If the departure-to-repair rates do not lie in either of the $\rho$ regimes, then the optimal repair policy (eager vs. lazy) depends on the relationship of the code parameters and the repair strategy (centralized or distributed).

### 5.2 Centralized vs. Distributed Repair

We now fix the departure rate $\lambda$ and repair rate $\mu$ to compare the repair cost of centralized vs. distributed repair per unit of time, as a function of the code parameters. Specifically, we determine relationships between $n, k, d$ and the code type (MSR vs. MBR) for which an optimal strategy can be derived. Our results are stated in the following two propositions.

**Proposition 4.** *For $d \leq \tau^* \leq n-1$, using MBR codes and distributed repair minimizes the average repair cost per unit of time, if $d > \frac{n+k-1}{3}$.*

*Proof.* Proof is provided in Appendix F. □

We now prove that if $\tau^*$ lies between $k$ and $d$, using MSR codes with centralized repair is optimal.

**Proposition 5.** *For $k \leq \tau^* < d$, the optimal repair strategy is given by centralized repair with MSR codes.*

*Proof.* Proof is provided in Appendix G. □

Although the above propositions show in which cases centralized repair is favorable over distributed repair, one also needs to be aware of the application involved. For example, when the devices belong to a single system/operator a centralized repair may be more likely to be used whereas when the system is heterogeneous with different type of devices a distributed repair may be more likely.

### 5.3 Mean Time to Data Loss for Periodic Repairs

We now examine the *Mean Time to Data Loss (MTTDL)* for the periodic threshold repair process. For our purposes, we consider that data is lost if the DSS transitions from state $\tau$ to state $\tau - 1$ instead of state $n$. That is, if a node leaves the system before repairs are completed when initiated at state $\tau$, the repair process is abandoned and the system eventually reaches state $k-1$, at which data is lost. In this case, the file $F$ is reinstated at the mobile nodes by a central entity. Note that when $\tau > k$ repairs could be re-initiated at state $\tau - 1$, because at least $k$ fragments remain available. We opted not to consider this option for the MTTDL calculation to capture the periodic nature of the threshold repair strategy. The MTTDL reflects the period of time at which the DSS oscillates between states $n$ and $\tau$. The time to reach state $k-1$ assuming no repairs are attempted after state $\tau$ is given by:

**Proposition 6.** *For a threshold-based repair strategy attempting regeneration at state $\tau$, the MTTDL is given by*

$$MTTDL = \sum_{i=1}^{\infty} \Big( \frac{iH_{n,\tau}}{\lambda} + \frac{i-1}{\mu} + \frac{H_{\tau,k-1}}{\lambda} \Big) (1-p)^{(i-1)} p, \quad (17)$$

*where $p = \frac{\tau\lambda}{\tau\lambda+\mu}$.*

*Proof.* Proof is provided in Appendix H. □

The MTTDL is a decreasing function of $\tau$. This is intuitive considering that the number of nodes that need to depart for reaching state $k-1$ increases with $\tau$. Moreover, the average time it takes to reach state $\tau$ from state $n$ increases with $\tau$. This indicates that the periodic repair of the DSS will on average last longer if a lazy repair strategy is adopted.

### 5.4 Numerical Examples

In this section, we validate our theoretical results be providing numerical examples. Fig. 5(a) shows $r(\tau)$ when $d > \frac{n+k-1}{3}$ and $\rho = 10^{-4}$ . According to Proposition 4, for this combination of code parameters, a distributed repair strategy with MBR codes (D-MBR) achieves the minimum $r(\tau)$ for all $d \leq \tau^* \leq n-1$. The minimum occurs at $\tau^* = d$. Moreover, according to Proposition 5, centralized MSR codes (C-MSR) minimize $r(\tau)$ for $k \leq \tau < d$. This is verified in all plots of Fig. 5, for which the cost is minimized by the C-MSR strategy when $\tau^* = k$, if $\tau < d$. In Fig. 5(b), we show $r(\tau)$ when $d < \frac{n+k-1}{3}$ and $\rho = 10^{-4}$. For this case, there is no one scheme with optimal cost for any value of $d \leq \tau \leq n-1$. For $\tau > 16$, D-MBR is optimal, whereas for $10 \leq \tau \leq 15$, C-MSR becomes optimal. C-MSR achieves the lowest overall cost at $\tau = k$.

We also studied the impact of $\rho$, when the code parameters are fixed to $(n = 30, k = 20, d = 25)$. Fig. 5(c) shows the average cost per unit of time ($r(\tau)$) when $\rho = 10^{-4}$. For this $\rho$ regime, a lazy repair strategy with $\tau^* = d$ minimizes $r(\tau)$, with D-MBR codes achieving the lowest cost. On the other hand, eager repair becomes optimal for any $\rho > \rho_{\text{high}}$. This is observed in Fig. 5(d), in which the value of $\rho$ has been increased to one. D-MBR codes still remain the optimal option, however, the optimal repair threshold is now shifted to $\tau^* = n-1$. Note that at the high $\rho$ regime, all codes exhibit the same behavior. The average cost per unit of time becomes a decreasing function of $\tau$.

Finally, on the right $y$-axis of the plots in Fig. 5, we show the MTTDL values for the given set of parameters. As expected, the MTTDL is an decreasing function of $\tau$ due to the corresponding increase in departure rate from state $\tau$ with the value of $\tau$. The MTTDL becomes impractical in the high $\rho$ regime, because nodes frequently leave area $\mathcal{A}$ before repairs can be completed.

## 6 Codes with Cooperative Repair

In the case of multiple node failures, regenerating the failed nodes individually is not optimal in terms of repair bandwidth. To regenerate multiple failed nodes more efficiently, the newcomers can also communicate with each other to lower the repair bandwidth, which is called *cooperative repair*. Specifically, the newcomer nodes communicate to not only the existing live nodes but also each of the other newcomers for regeneration. In this section, we analyze examples of such codes and their performance for the Markov-model that is described earlier.

### 6.1 Cooperative Regenerating Codes

When multiple nodes are to be repaired simultaneously, in addition to contacting live nodes and downloading symbols from those, newcomers can also communicate between each other to complete the recovery process. Formally, assume that $t$ nodes are to be repaired. Each of the $t$ newcomer nodes can contact $d$ live nodes and download $\beta$ symbols as well as download $\beta'$ from each other. In this scenario, the repair bandwidth can be calculated as $\gamma = d\beta + (t-1)\beta'$. Such codes are studied in [25] (referred to also as coordinated regenerating codes) and the tradeoff between per

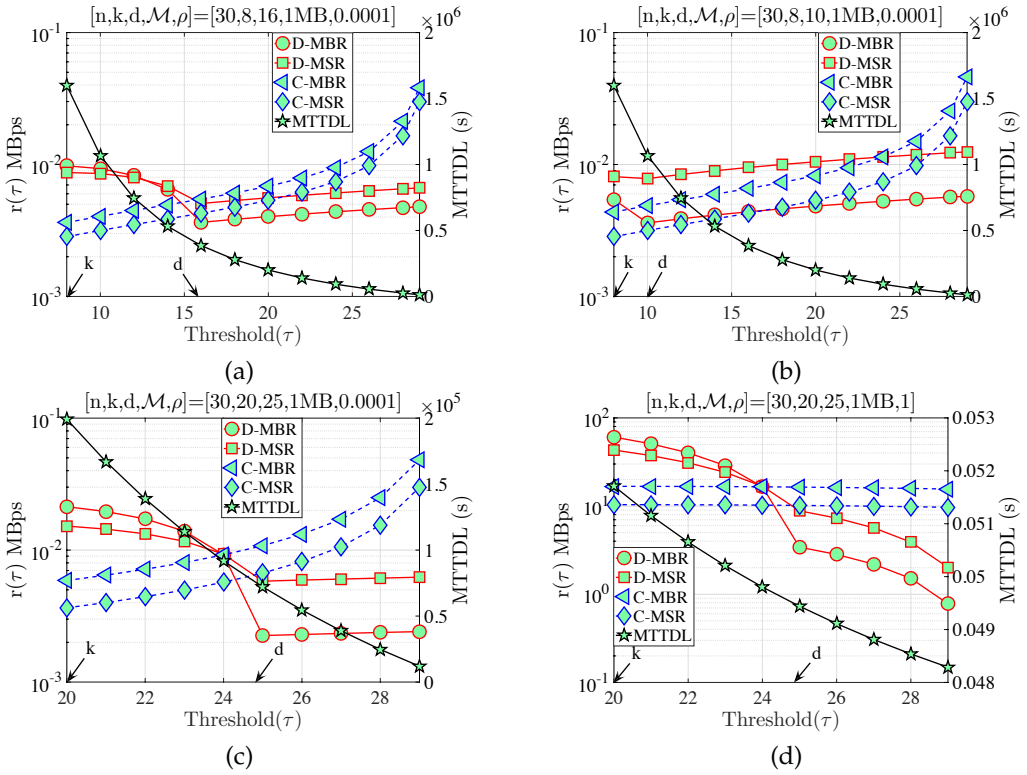



Fig. 5: Cost $r(\tau)$ vs. repair threshold ($\tau$) for: (a) $d > \frac{n+k-1}{3}$, (b) $d < \frac{n+k-1}{3}$ >, (c) $\rho = 10^{-4}$, (d) $\rho = 1$.

node storage $\alpha$ and repair bandwidth $\gamma$ is analyzed. Two ends of tradeoff curve is named Minimum Storage Cooperative Regenerating (MSCR) and Minimum Bandwidth Cooperative Regenerating (MBCR). Accordingly, operating points are given as follows:

$$(\alpha_{\text{MSCR}}, \beta_{\text{MSCR}}, \beta'_{\text{MSCR}}) = \Big(\frac{\mathcal{M}}{k}, \frac{\mathcal{M}}{k\,(d-k+t)}, \frac{\mathcal{M}}{k\,(d-k+t)}\Big). \tag{18}$$

For MSCR codes, $\gamma_{\text{MSCR}} = d\beta_{\text{MSCR}} + (t-1)\beta'_{\text{MSCR}} = \frac{\mathcal{M}(d+t-1)}{k(d-k+t)} \geq \frac{\mathcal{M}}{k}$, where the per-node storage is smaller than the repair bandwidth. On the other hand, MBCR codes provide the minimum repair bandwidth which operates at:

$$(\alpha_{\text{MBCR}}, \beta_{\text{MBCR}}, \beta'_{\text{MBCR}}) = \Big(\frac{(2d+t-1)\mathcal{M}}{k\,(2d-k+t)}, \frac{2\mathcal{M}}{k\,(2d-k+t)}, \frac{\mathcal{M}}{k\,(2d-k+t)}\Big). \tag{19}$$

Note that for MBCR codes, we have $\alpha_{\text{MBCR}} = \gamma_{\text{MBCR}} = d\beta_{\text{MBCR}} + (t-1)\beta'_{\text{MBCR}}$.

We define $r_t(\tau)$ as the average repair cost for a system with repair threshold $\tau$ under cooperative repair using groups of nodes of size $t$ (similarly $c_t(\tau)$ for the cost). Since $n-\tau$ nodes need to be repaired, any cooperative regenerating codes with $t$ such that $t|n-\tau$ can be used in practice. In the following proposition, we compare the performance of cooperative regenerating codes at all possible $t$ values to find the value of $t$ that minimizes the average repair cost.

**Proposition 7.** *The average repair cost of cooperative repair is a monotonically decreasing function of the cooperation group size $t$. That is, for two cooperation groups $t_1$ and $t_2$, with $t_1|n-\tau$ and $t_2|n-\tau$ and $t_1 < t_2$, it follows that $r_{t_1}(\tau) > r_{t_2}(\tau)$.*

*Proof.* Proof is provided in Appendix I. □

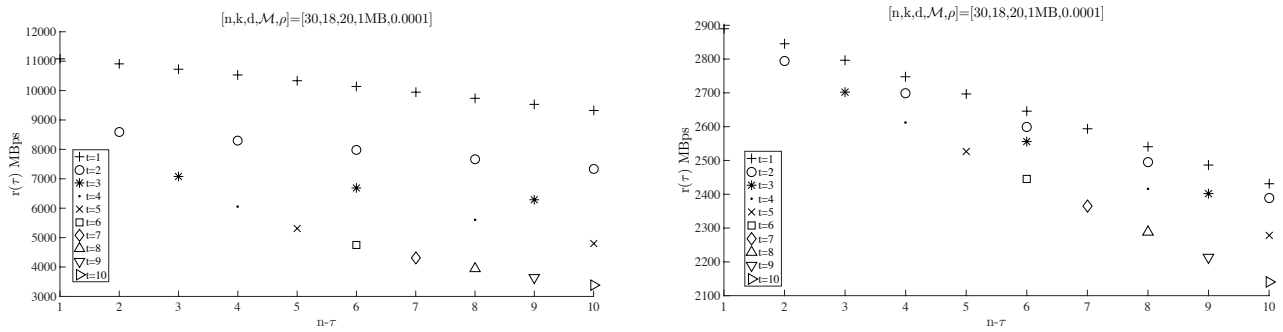
**Remark 1.** *As a result of the above proposition, one can minimize the average repair cost by performing cooperative repairs with $t = n-\tau$. In other words, for the $n-\tau$ nodes that are to be repaired, the optimal cooperative regenerating code is with $n-\tau$, all nodes should cooperate at the same time.*

**Remark 2.** *The above proposition does not take into account the inherent cost of coordinating the symbol exchange during the cooperative repair process. This cost depends on the specific implementation details of the protocol that facilitates the cooperation, the network topology (one hop, vs. multihop) and the communication mode (broadcast vs. multicast, vs. unicast).*

In Fig. 6, we show the average repair cost for cooperative regenerating codes at all possible $t$ values for a given $n-\tau$. We observe that for the same $n-\tau$, if $t_1 < t_2$, then $r_{t_1}(\tau) > r_{t_2}(\tau)$ and the minimum is achieved when $t = n-\tau$.

In the remaining of this section, we suppress the subindex $t$ from $r_t(\tau)$ since we established that $t = n-\tau$ minimizes the cost, i.e., $r(\tau) = r_{n-\tau}(\tau)$ and $c(\tau) = c_{n-\tau}(\tau)$. However, we may still need to distinguish $\gamma$ and $\alpha$ values under different cooperative regenerating codes. We denote the per-node storage for cooperative repairs with $n-\tau$, which results in the minimum average repair cost for the system with threshold $\tau$, by $\alpha_\tau$. Similarly we use $\gamma_\tau$ to denote the repair bandwidth at $\tau$.

Note that, cooperative regenerating codes, it is required that $d+n-\tau \leq n$. In other words, there should be at least $d$ live nodes when repairs are initiated. Otherwise, it is not possible to regenerate $n-\tau$ nodes from d live nodes. Henceforth, the repair cost $c(\tau)$ and the repair cost per time for cooperative codes are as follows:




(a) (b)

Fig. 6: Cost $r(\tau)$ vs. number of nodes to be repaired $(n-\tau)$ for: (a) MSCR, (b) MBCR.

$$c(\tau)=\gamma_\tau(n-\tau),\quad r(\tau)=\frac{c(\tau)}{\mathrm{E}[\Delta]}=\frac{\lambda\mu(\gamma_\tau(n-\tau))}{\mu H_{n,\tau}+\lambda},\ \text{if } \tau\geq d. \tag{20}$$

The minimum $r(\tau)$ with respect to $\tau$ does not have a closed-form analytical expression. In Section 6.3, we present numerical results to study the change of $r(\tau)$ with $\tau$ and determine the optimal repair threshold $\tau$ that minimizes $r(\tau)$ for distributed cooperative repair.

### 6.2 Centralized Repair of Multiple Node Failures

Centralized repair of multiple node repairs is introduced in [28]. Using this model, a dedicated node among the $t$ newcomers downloads $\beta$ from any $d$ live nodes such that it can repair multiple node failures of size $t$. Such codes can be used in the centralized repair process that is proposed in Section 4 when we set $t=n-\tau$. Rawat *et al.* in [28] characterize the tradeoff between per-node storage and repair bandwidth for this centralized repair model. Accordingly, the following operation points are derived for minimum storage multi-node regeneration (MSMR) and minimum bandwidth multi-node regeneration (MBMR):

$$(\alpha_{\text{MSMR}},\gamma_{\text{MSMR}})=\left(\frac{\mathcal{M}}{k},\frac{\mathcal{M}d(n-\tau)}{k\left(d-k+n-\tau\right)}\right). \tag{21}$$

Let $k \bmod (n-\tau)=b$. If $H_b\geq \binom{\beta}{n-\tau}\left[b(\frac{2d+n-\tau-1}{2})-\binom{b}{2}\right]$ (where $H_b$ denotes entropy of information stored on $b$ nodes), then

$$(\alpha_{\text{MBMR}},\gamma_{\text{MBMR}})=\left(\frac{\mathcal{M}2d}{k\left(2d-k+n-\tau\right)},\frac{\mathcal{M}2d(n-\tau)}{k\left(2d-k+n-\tau\right)}\right). \tag{22}$$

Under centralized repair model discussed here, a dedicated node first downloads $\gamma=d\beta$ and then distributes $\alpha$ to remaining $n-\tau-1$. The difference between these codes and the earlier centralized repair in Section 4.2 is that the dedicated node may not need to download the whole file. Therefore, we have the following repair cost

$$c_C(\tau)=\gamma+\alpha(n-\tau-1), \tag{23}$$

from which one can obtain

$$r_C(\tau)=\frac{c_C(\tau)}{\mathrm{E}[\Delta]}=\frac{\lambda\mu(\gamma+\alpha(n-\tau-1))}{\mu H_{n,\tau}+\lambda}. \tag{24}$$

In order to find the optimal threshold that minimizes the average repair cost, we need to find the minimum value of $r_C(\tau)$. We can replace $H_{n,\tau}$ with its approximation, $\ln(\frac{n}{\tau})$, and take the derivative with respect to $\tau$. Note that both $\gamma$ and $\alpha$ depend on $\tau$ and there is no tractable analytical solution for $\tau$ that minimizes $r_C(\tau)$. However, we can still analyze (24) numerically with respect to $\tau$ and observe the optimal threshold from numerical results.

**Remark 3.** *In this section, we analyzed different cooperative codes that are suitable for mobile clouds. For both scenarios, the problem of finding $\tau$ for which $r(\tau)$ is minimized does not have a closed-form solution due to the dependence of $\alpha$ and $\gamma$ on $\tau$. We, therefore, resort to the numerical analysis of the optimal threshold. The numerical results are used to illustrate the inherent tradeoffs between coding parameters and the repair bandwidth. Note that, for the regenerating codes that were analyzed in Section 5, $\alpha$ and $\gamma$ do not depend on the threshold $\tau$.*

### 6.3 Numerical Results

We study the performance of cooperative codes with $n=30, d=25$ and $k=19$ under different $\rho$ regimes. In Fig. 7, we compare the codes studied in Sections 4 and 6 for different $\rho$ regimes. We first compare regenerating codes vs. cooperative regenerating codes for the distributed repair scenario. Note that we focus only on $d\leq\tau\leq n-1$ because at least $d$ live nodes must exist for cooperation (see Section 6.1). We observe that cooperative regenerating codes always have lower cost than regenerating codes for all values of $\rho$. Additionally, the gap between the cost of D-MBR and D-MBCR is much smaller than the gap between the cost of D-MSR and D-MSCR. Furthermore, we can observe two opposing regimes: In Fig. 7(a), the optimal cost is at $\tau=d$, whereas in Fig. 7(b), the cost is minimized at $\tau=n-1$. We also compare the centralized regenerating codes to the centralized repair of multiple node departures. As expected, since in the latter scheme one does not need the whole file for file reconstruction at the dedicated node, centralized repair of multiple node departures results in lower repair cost. At $\tau=n-1$, in Fig. 7(d) average repair cost is minimized for all schemes, on the other hand we observe different optimum $\tau$ values for different coding schemes in Fig. 7(c). Finally, we compare all schemes in Fig. 7(e)-(f) for different values of $\rho$ for completeness. It is observed that the centralized repair of multiple node departures model discussed in this section approaches to the distributed repair model in Section 4 as $\tau$ approaches $n$ and diverges from the centralized repair model in Section 4. The reason for

this behavior is that the dedicated node does not need to download the whole file now (as opposed to the centralized repair model in Section 4, which incurs high average repair cost for large $\tau$).

## 7 A Repair Process Analogous to the number of repaired nodes

In the analysis presented in Sections 5-6, we have assumed that the repair time is exponentially distributed with parameter $\mu$, irrespective of the number of nodes to be repaired. This model is mathematically tractable and we are able to find analytical results on the optimal repair threshold. In this section, we consider a revised model in which the repair time is analogous to the number of nodes that need to be repaired. Specifically, we model the repair process as the maximum of $n-\tau$ exponential random variables, each with rate $\mu$. In other words, when the repair process is initiated, one can consider starting $n-\tau$ exponential clocks, each with rate $\mu$. The repair process ends when all the clocks end. We note that the maximum value of such clocks is not exponentially distributed (as opposed to the minimum of such clocks), however, its expected value is known, which is enough for the purpose of finding average repair cost. Let $T_i^r$ denote the repair time of $i^{\text{th}}$ newcomer node, then we have the following [34]

$$\mathrm{E}[\max(T_1^r,\ldots,T_{n-\tau}^r)] = \sum_{i=1}^{n-\tau}\frac{1}{i\mu} = \frac{H_{n-\tau,0}}{\mu}. \tag{25}$$

The expected time between two instances of fully operational system with $n$ live nodes is given by

$$\mathrm{E}[\Delta] = \frac{H_{n,\tau}}{\lambda} + \frac{H_{n-\tau,0}}{\mu}, \tag{26}$$

where the first term is the expected time from state $n$ to state $\tau$, and the second term is the expected time from state $\tau$ back to state $n$. Accordingly, the resulting average repair cost for the distributed repair case is

$$r_D(\tau) = \frac{c_D(\tau)}{\mathrm{E}[\Delta]} = \begin{cases} \frac{\lambda\mu(k\alpha(d-\tau)+\gamma(n-d))}{\mu H_{n,\tau}+\lambda H_{n-\tau,0}}, & \text{if } \tau < d \\ \frac{\lambda\mu(\gamma(n-\tau))}{\mu H_{n,\tau}+\lambda H_{n-\tau,0}}, & \text{if } \tau \geq d. \end{cases} \tag{27}$$

Note that for centralized repair, a dedicated newcomer node first downloads the file, and then distributes symbols to the remaining $n-\tau-1$ newcomer nodes. Therefore, the expected repair time is given by $\frac{1}{\mu}+\frac{H_{n-\tau-1}}{\mu}$, where we have first one clock with exponential rate of $\mu$, followed by a maximum of $n-\tau-1$ clocks with rate $\mu$. Accordingly,

$$r_C(\tau) = \frac{c_C(\tau)}{\mathrm{E}[\Delta]} = \frac{\lambda\mu\alpha(k+n-\tau-1)}{\mu H_{n,\tau}+\lambda(1+H_{n-\tau-1,0})}. \tag{28}$$

The optimal $\tau$ which minimizes the above equation is difficult to track due to the complexity of the formula (due to $\lambda H$ in the denominator). Instead, we perform numerical analysis of the average repair cost with respect to the threshold later in this section.

In the context of this model, we next focus on MTTDL. Note that, if we start $(n-\tau)$ clocks, the probability that no data loss occurs within a cycle, denoted by $1-p$, can be calculated as $1-p = \Pr(T_{\tau-1} > T_i^r, \forall i \in 1,\ldots,n-\tau)$. In other words, the exponential random variable with rate $\tau\lambda$ should be greater than all $n-\tau$ exponential random variables with rate $\mu$. Since we have i.i.d. exponential random variables, $1-p = (\Pr(T_{\tau-1} > T_1^r))^{(n-\tau)} = (\frac{\mu}{\tau\lambda+\mu})^{(n-\tau)}$. Accordingly, we have

$$\text{MTTDL} = \sum_{i=1}^{\infty}\left(\frac{iH_{n,\tau}+H_{\tau,k-1}}{\lambda}+\frac{(i-1)H_{n-\tau,0}}{\mu}\right)p(1-p)^{(i-1)} \tag{29}$$

where $p = 1-(\frac{\mu}{\tau\lambda+\mu})^{(n-\tau)}$. Note that the above equation is for the calculation of MTTDL for distributed repair. For centralized repair, the random variable with rate $\tau\lambda$ should be larger than sum of two exponential random variables with rate $\mu$, which is a gamma distribution since first a dedicated node downloads the whole file and then it repairs the other nodes. Henceforth, we did not perform MTTDL analysis for centralized repair. [2]

In Fig. 8, we compare how the revised model affects the average repair cost relative to the simplified repair model used in Section 4, in which all nodes are repaired under the same clock. We denote the values calculated within this section by appending $\tau$ at the end, i.e., D-MBR-$\tau$, to specify that the repair process that takes into account the number of nodes to be repaired (i.e., $n-\tau$). It can be observed that changing the model does not affect the behavior of the $r(\tau)$ curves substantially. We observe in the modified model that average cost is decreased slightly. Even though we have the same costs in both cases, the expected times to complete the repair process are different. Specifically, in Sections 4 and 5, it takes $\frac{1}{\mu}$ time to finish the repair process. On the other hand, in the modified model, we change this value to maximum of $n-\tau$ exponential random variables, each with mean $\frac{1}{\mu}$ and this maximum value is larger than $\frac{1}{\mu}$ (unless $\tau = n-1$). Therefore, it takes longer to complete repairs in the modified model, which results in smaller values of average repair cost. On the other hand, in Fig. 9, a different behavior is observed for the MTTDL. In the low $\rho$ regime, the MTTDL decreases with $\tau$ for both single clock and the maximum of multiple clock models. However, in the high $\rho$ regime, the increase in $\tau$ decreases MTTDL for the single clock model, whereas the MTTDL for the multiple clocks model increases with $\tau$. This is because in (29), $p$ converges to 1 as we decrease $\tau$ in the high $\rho$ regime, which reduces (29). On the other hand, in the low $\rho$ regime, $p$ converges to zero as we increase $\tau$, which reduces (29). Finally, the model discussed in this section results in lower MTTDL values compared to the previous one.

## 8 Node Departures During Repair

Up to this point, we have assumed that once the repair process is started, no live nodes depart from $\mathcal{A}$ until the repair is completed. In this section, we analyze the case in which additional node departures are allowed within repair process. Fig. 10 shows the revised CTMC model that accounts for departures during the repair process. The chain consists of two sets of states. In the first row, states represent the phase where nodes depart but no repairs are initiated. Once state $\tau$ is reached, repairs are initiated and the chain transitions into the second row of states where departures

2. The resulting equation is not in compact form, thus we omit this analysis in this text.

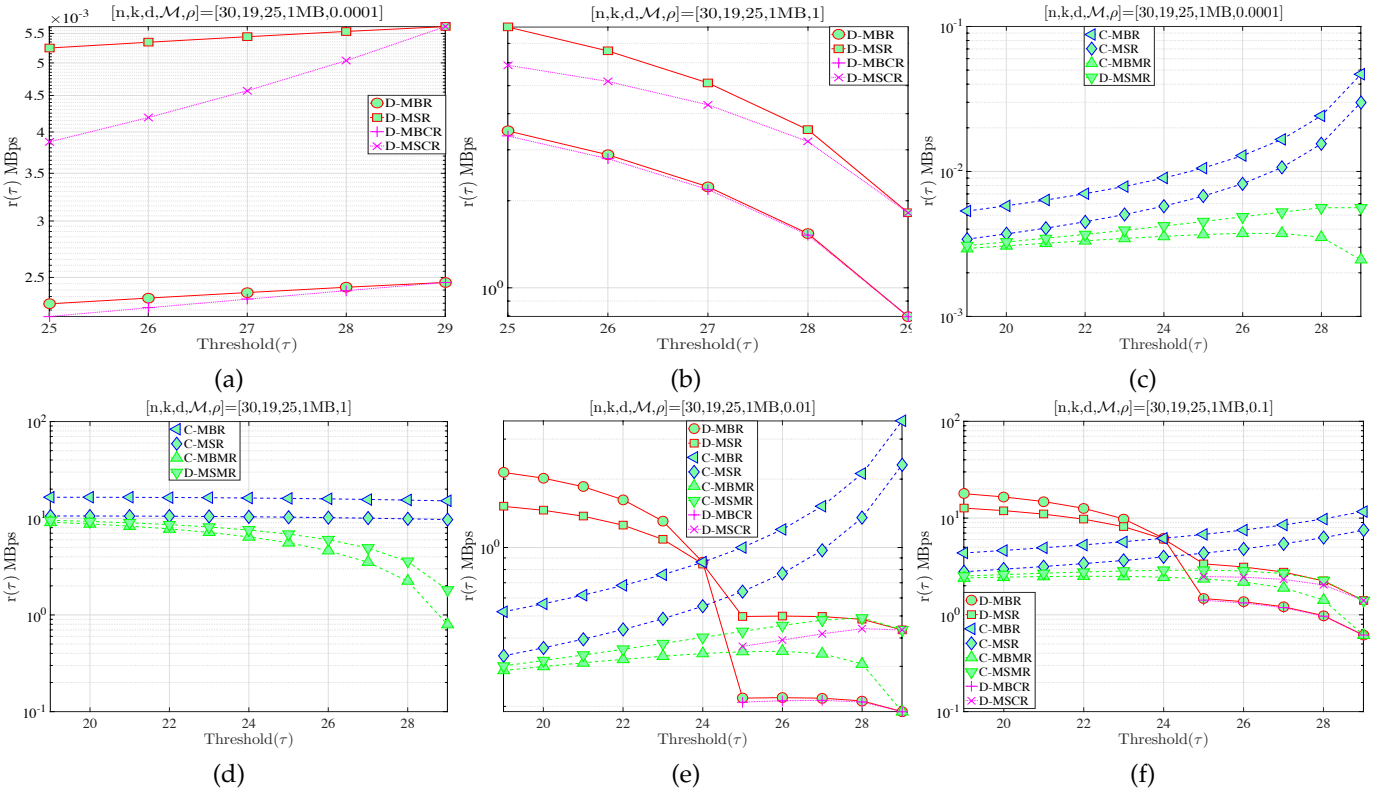



Fig. 7: Cost $r(\tau)$ vs. repair threshold $(\tau)$ for: (a) distributed regenerating codes vs. cooperative regenerating codes when $\rho = 0.0001$, (b) distributed regenerating codes vs. cooperative regenerating codes when $\rho = 1$ (c) centralized regenerating codes vs. centralized repair of multiple node failures when $\rho = 0.0001$, (d) centralized regenerating codes vs. centralized repair of multiple node failures when $\rho = 1$, (e) all schemes when $\rho = 0.01$, (f) all schemes when $\rho = 0.1$.

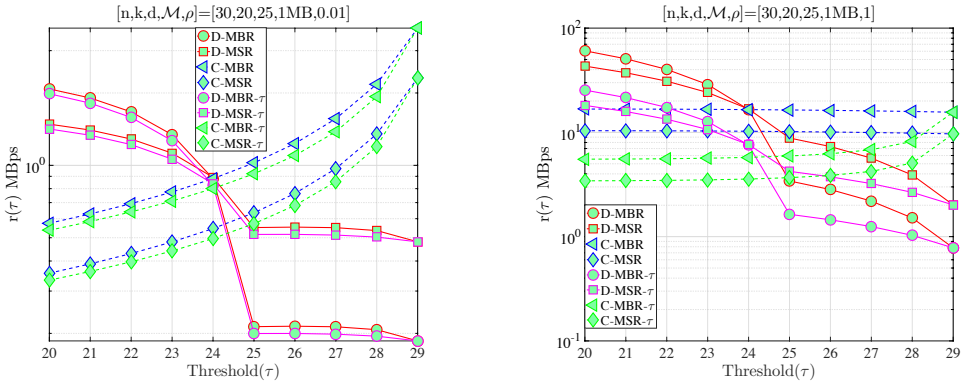



Fig. 8: Cost $r(\tau)$ vs repair threshold $(\tau)$ for: (a) $\rho = 0.01$, (b) $\rho = 1$.

may occur while repairs are performed. We focus on the case where no data loss occurs as we are interested in the system dynamics, while the system remains operational.

Once the repair process is initiated, $n-\tau$ nodes are to be repaired. Since we assume an exponential distribution for repair times (with rate $\mu$), a newcomer is repaired after the minimum of the $n-\tau$ exponential random variables, which is also an exponential random variable with rate $(n-\tau)\mu$. Due to the memoryless property of the exponential distribution, we can perform the same procedure for the second repair and so on. The corresponding rates are depicted on the second row of states of Fig. 10. Note that if no departures occur during repair, the expected repair time would be the same as in the model of Section 7, that is, the maximum of $n-\tau$ exponential random variables with rate $\mu$.

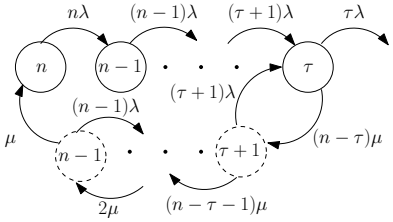
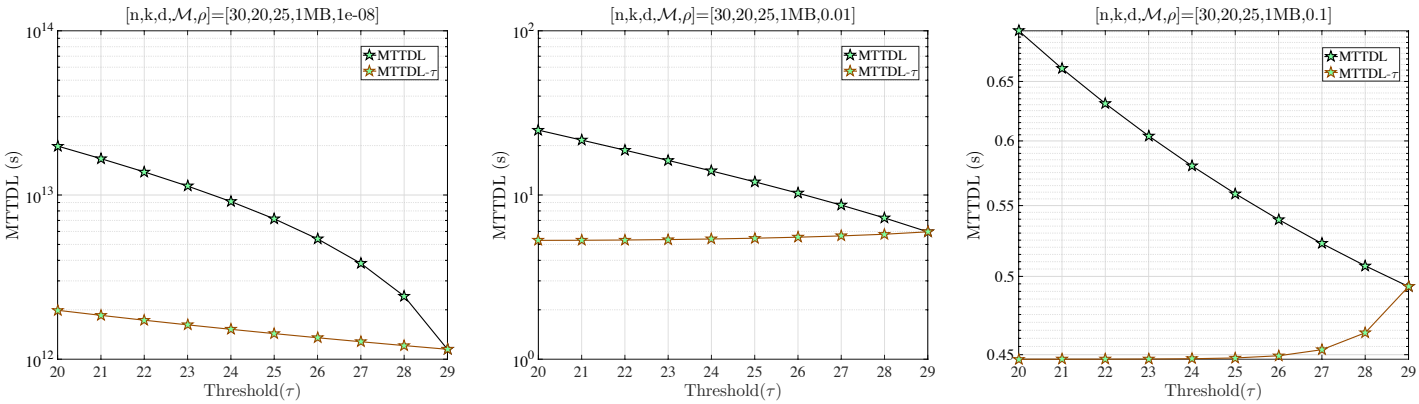



Fig. 10: Markov chain for a threshold-based file maintenance.




(a) (b) (c)

Fig. 9: Mean time to data loss vs repair threshold $(\tau)$ for: (a) $\rho = 0.004$, (b) $\rho = 0.1$, (c) $\rho = 1$.

In such a model, the expected number of node repairs is larger than $n - \tau$ due to the possible additional departures during the repair process. In the section, we analyze the system under the condition that the CTMC always chooses the transition $\tau \rightarrow \tau + 1$ when in the lower set of states so that all nodes are repaired eventually. Otherwise, the system would suffer from data loss. To find the average repair cost, we are interested in two statistics of the CTMC once it reaches state $\tau$ for the first time: i) the total number of lower arc transitions before reaching state $n$, i.e., $\tau \rightarrow \tau+1, n-2 \rightarrow n-1$, which will determine the number of node repairs, ii) the expected total time for reaching state $n$ from state $\tau$. For simplicity, the states $n-1, \ldots, \tau+1$ can be ignored for now since they do not affect any of these two statistics. We are also interested in the number of times the chain revisits state $\tau$ before reaching state $n$ since that will determine the probability of no data loss.

### 8.1 Total number of revisits to state $\tau$

Since we are interested in the case where no data loss occurs, we need to find the number of times the chain revisits state $\tau$. This is equivalent to transitioning from state $\tau$ to $\tau + 1$ at each revisit to state $\tau$, we need the transition $\tau \rightarrow \tau + 1$ at each revisit to state $\tau$, which occurs with probability $\frac{(n-\tau)\mu}{\tau\lambda+(n-\tau)\mu}$. For the transitions in the lower row of the CTMC, denote by $X_j^r$, the total number of revisits to state $\tau$ before reaching state $n$ at state $j$. Then, we can state the following balance equations.

$$X_{n-1}^r = \frac{X_{n-2}^r(n-1)\lambda}{(n-1)\lambda+\mu}, \tag{30}$$

$$X_{n-2}^r = \frac{X_{n-1}^r 2\mu + X_{n-3}^r(n-2)\lambda}{(n-2)\lambda+2\mu}, \tag{31}$$

$$\vdots$$

$$X_{\tau+2}^r = \frac{X_{\tau+3}^r(n-\tau-2)\mu + X_{\tau+1}^r(\tau+2)\lambda}{(\tau+2)\lambda+(n-\tau-2)\mu}, \tag{32}$$

$$X_{\tau+1}^r = \frac{X_{\tau+2}^r(n-\tau-1)\mu + (1+X_\tau^r)(\tau+1)\lambda}{(\tau+1)\lambda+(n-\tau-1)\mu}, \tag{33}$$

$$X_\tau^r = X_{\tau+1}^r. \tag{34}$$

The set of equations (30)-(34) can be recursively solved for $X_\tau^r$. Then, the probability that no data loss occurs is $(\frac{(n-\tau)\mu}{\tau\lambda+(n-\tau)\mu})^{1+X_\tau^r}$ for one cycle of node repairs.

### 8.2 Total number of lower arc transitions

For the transitions of the lower row of the CTMC, denote by $X_j^l$, the total number of lower arc transitions at state $j$, then we have the following equations.

$$X_{n-1}^l = \frac{\mu + X_{n-2}^l(n-1)\lambda}{(n-1)\lambda+\mu}, \tag{35}$$

$$X_{n-2}^l = \frac{(1+X_{n-1}^l)2\mu + X_{n-3}^l(n-2)\lambda}{(n-2)\lambda+2\mu}, \tag{36}$$

$$\vdots$$

$$X_{\tau+1}^l = \frac{(1+X_{\tau+2}^l)(n-\tau-1)\mu + X_\tau^l(\tau+1)\lambda}{(\tau+1)\lambda+(n-\tau-1)\mu}, \tag{37}$$

$$X_\tau^l = 1 + X_{\tau+1}^l. \tag{38}$$

Finding the total number of lower arc transitions is not enough to calculate the average repair cost per time since not all repairs have the same cost in some repair strategies. That is, if $\tau < d$, then some nodes are repaired by downloading $k\alpha$ symbols, whereas the remaining nodes are repaired by downloading $d\beta$. To find the number of lower arc transitions which occur between states $\tau$ and $d$ (which are repaired by downloading $k\alpha$), we denote by $Y_j^l$ the number of lower arc transitions between states $\tau$ and $d$ at state $j$. Then,

$$Y_{n-1}^l = \frac{Y_{n-2}^l(n-1)\lambda}{(n-1)\lambda+\mu} \tag{39}$$

$$Y_{n-2}^l = \frac{Y_{n-1}^l 2\mu + Y_{n-3}^l(n-2)\lambda}{(n-2)\lambda+2\mu}, \tag{40}$$

$$\vdots$$

$$Y_d^l = \frac{Y_{d+1}^l(n-d)\mu + Y_{d-1}^l d\lambda}{d\lambda+(n-d)\mu}, \tag{41}$$

$$Y_{d-1}^l = \frac{(1+Y_d^l)(n-d+1)\mu + Y_{d-2}^l(d-1)\lambda}{(d-1)\lambda+(n-d+1)\mu}, \tag{42}$$

$$\vdots$$

$$Y_{\tau+1}^l = \frac{(1+Y_{\tau+2}^l)(n-\tau-1)\mu + Y_\tau^l(\tau+1)\lambda}{(\tau+1)\lambda+(n-\tau-1)\mu}, \tag{43}$$

$$Y_\tau^l = 1 + Y_{\tau+1}^l. \tag{44}$$

If $\tau \geq d$, there is no need to find $Y_\tau^l$ since all nodes download $d\beta$ and there are $X_\tau^l$ node repairs in total whereas if $\tau < d$, then $Y_\tau^l$ repairs are performed by downloading $k\alpha$ symbols and $X_\tau^l - Y_\tau^l$ node repairs are performed by downloading $d\beta$ symbols.

### 8.3 Expected total time before reaching state $n$

For the transitions of the lower row of the CTMC, denote by $X_j^t$, the time it takes to reach state $n$ from state $j$. Then, the balance equations for the CTMC for $X_j^t$ can be written as

$$X_{n-1}^t = \frac{1+X_{n-2}^t(n-1)\lambda}{(n-1)\lambda+\mu}, \quad (45)$$

$$X_{n-2}^t = \frac{1+X_{n-1}^t 2\mu+X_{n-3}^t(n-2)\lambda}{(n-2)\lambda+2\mu}, \quad (46)$$

$$\vdots$$

$$X_{\tau+1}^t = \frac{1+X_{\tau+2}^t(n-\tau-1)\mu+X_{\tau}^t(\tau+1)\lambda}{(\tau+1)\lambda+(n-\tau-1)\mu}, \quad (47)$$

$$X_{\tau}^t = \frac{1}{(n-\tau)\mu} + X_{\tau+1}^t. \quad (48)$$

Using (45)-(48), we can solve for $X_\tau^t$ which is the time it takes to repair to a fully operational system with $n$ live nodes from state $\tau$, when departures occur during the repair process. The initial state for the system is $n$ and therefore the time to revisit state $n$ is $X_\tau^t + \sum_{i=\tau+1}^{n} \frac{1}{i\lambda}$.

This yields an average cost per time equal to

$$r_D(\tau|\text{no data loss occurs}) = \frac{c_D(\tau|\text{no data loss occurs})}{\text{E}[\Delta|\text{no data loss occurs}]} = \begin{cases} \frac{Y_\tau^l k\alpha+(X_\tau^l-Y_\tau^l)d\beta}{X_\tau^t+\sum_{i=\tau+1}^{n}\frac{1}{i\lambda}}, & \text{if } \tau < d \\ \frac{X_\tau^l d\beta}{X_\tau^t+\sum_{i=\tau+1}^{n}\frac{1}{i\lambda}}, & \text{if } \tau \geq d. \end{cases} \quad (49)$$

and a probability of no data loss equal to $(\frac{(n-\tau)\mu}{\tau\lambda+(n-\tau)\mu})^{1+X_\tau^r}$.

### 8.4 Numerical Results

We have performed simulations to verify our findings. The simulation of Markov chain for the threshold-based file maintenance is performed with MATLAB. We initialize the Markov Chain at state $n$ and we arrive at state $\tau$ since there are only jumps to right as the repair process is not initiated yet. After we reach state $\tau$, the repair process is initiated and we only simulate the case where no data loss occurs. In order to do that, we enforce a jump from state $\tau$ to $\tau+1$ (in the lower arc). During repair process, Markov chain may transition to state $\tau$ (since some nodes may depart the system while the repairs are not finished) but every time it reaches that state, we enforce the jump we discussed before to ensure that there is no data loss. A simulation is finished when the state of the chain transitions to the state $n$. However, we repeat the same simulation one million times and take the averages of the statistics and report them. In the following example, we examine the case where $n = 30$, $d = 27$, $k = 20$, $\mu = 10$. Different $\lambda$ values are used, $\lambda = [0.1, 0.2, 0.4]$, as well as different $\tau$ values, $\tau = [25, 27]$.

The simulation results are the average of one million simulations and they are presented in Tables 2-5. Each table entry notes the value obtained from the simulation or the value obtained by analytically evaluation the average repair cost via (49): the first shows the simulation results (S) and the other represents the analytical result (A) as shown before. As expected, increasing $\lambda$ results in more revisits to state $\tau$ due to an increase in the node departure rate. Furthermore, we see that expected time before reaching state $n$ decreases as we increase $\lambda$ for both cases of $\tau = 25$ and $\tau = 27$. For both cases, we observe that the number of nodes repaired by downloading $d\beta$ remains the same for a given value of $\lambda$. This is because the critical number of live nodes for a node to be repaired by downloading $d\beta$ symbols is at $d = 27$, since if there are less than $d$ live nodes, then a node must be repaired by downloading $k\alpha$ symbols. In other words, once we reach state $d$ (in the lower row) of the CTMC, we count the number of lower arc transitions between the states $d$ and $n$ to calculate the number of node repairs by downloading $d\beta$ symbols, which remains the same for $\tau = 25$ and $\tau = 27$, since $\tau$ is not between $d$ and $n$. Finally, the number of nodes repaired by downloading $k\alpha$ increases with $\lambda$ as expected. We can observe that simulation results verify our analytical findings.

In Fig. 11, we compare our previous schemes, namely the distributed repair model in Section 4 and the model in Section 7, with the repair model discussed in this section, which is depicted in the figure with D-MBR-F and D-MSR-F to specify that these codes allow failures within repair process. In the low $\rho$ regime, we observe that there is almost no difference between the performance of models in both MSR and MBR cases. This is because for low $\rho$, the expected number of additional departures during repair is low and the expected time is dominated by terms with $\lambda$ (the upper transition in the CTMC model of Fig. 10, which is the same as the CTMC model of Fig. 4). In the high $\rho$ regime, differences are observed in the expected average cost per time. Interestingly, when $\rho = 0.1$, it can be observed that for $\tau \geq d-1$, D-MSR-F and D-MBR-F have the highest cost respectively for MSR and MBR cases, whereas for $\tau < d-1$, they are in between the D-MSR and D-MBR. As we keep increasing $\rho$, we observe that D-MSR-F and D-MBR-F becomes even more costly compared to other schemes. In all cases, our model in Section 7 has the lowest $r(\tau)$ respectively for MSR and MBR cases. These observations validate that our model in Section 4 (that does not have dependency of repair process on the number of nodes to be repaired and neglect failures during repair) can be utilized as an approximate model for the models we consider later in the text in the low $\rho$ regime. Therefore, in the low $\rho$ regime, all the optimal threshold statements for the former model (i.e., Proposition 1, Proposition 2) also hold for the model considered in Sections 7 and 8. Note that, the low $\rho$ regime is the only interesting case for mobile cloud storage. At high $\lambda$, the average repair cost and MTTDL performance become prohibitively high and low, respectively, for the system to be viable.

## 9 Conclusion and Future Work

We analyzed threshold-based repair strategies for maintaining files in dynamic DSS with emphasis on mobile cloud storage systems. We derived the optimal repair thresholds for both distributed and centralized repair schemes under fragment regeneration and/or reconstruction. Our results showed that optimal thresholds are dependent on system configurations, the underlying code parameters and departure-to-repair rate ratio. For high departure-to-repair scenarios, eager repair minimizes the average repair cost per unit of time. Under low departure-to-repair ratio, lazy

TABLE 2: Total number of revisits to state $\tau$

| | | $\lambda$ | | | | | |
|---|---|---|---|---|---|---|---|
| | | 0.1 | | 0.2 | | 0.4 | |
| | | S | A | S | A | S | A |
| $\tau$ | 25 | 1.0718 | 1.0719 | 1.1633 | 1.1638 | 1.4660 | 1.4668 |
| | 27 | 1.1806 | 1.1806 | 1.4439 | 1.4424 | 2.2118 | 2.2096 |

TABLE 3: Expected total time before reaching state $n$

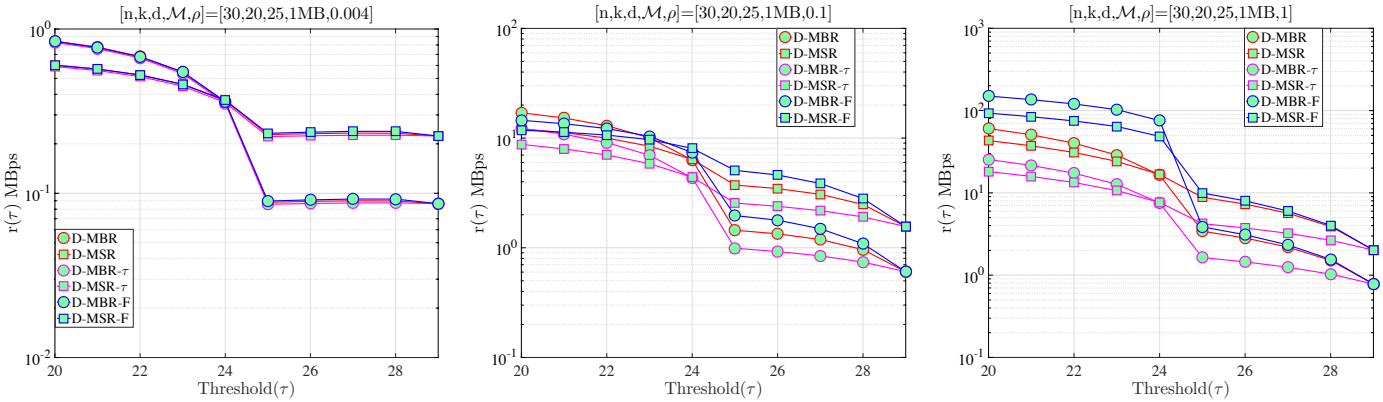
| | | $\lambda$ | | | | | |
|---|---|---|---|---|---|---|---|
| | | 0.1 | | 0.2 | | 0.4 | |
| | | S | A | S | A | S | A |
| $\tau$ | 25 | 2.0438 | 2.0432 | 1.1770 | 1.1770 | 0.8040 | 0.8034 |
| | 27 | 1.2404 | 1.2392 | 0.7458 | 0.7447 | 0.5402 | 0.5405 |

TABLE 4: Number of node repairs by downloading $d\beta$

| | | $\lambda$ | | | | | |
|---|---|---|---|---|---|---|---|
| | | 0.1 | | 0.2 | | 0.4 | |
| | | S | A | S | A | S | A |
| $\tau$ | 25 | 3.4703 | 3.4706 | 4.0263 | 4.0224 | 5.3702 | 5.3696 |
| | 27 | 3.4715 | 3.4706 | 4.0237 | 4.0224 | 5.3727 | 5.3696 |

TABLE 5: Number of node repairs by downloading $k\alpha$

| | | $\lambda$ | | | | | |
|---|---|---|---|---|---|---|---|
| | | 0.1 | | 0.2 | | 0.4 | |
| | | S | A | S | A | S | A |
| $\tau$ | 25 | 2.1784 | 2.1782 | 2.4228 | 2.4234 | 3.2620 | 3.2623 |
| | 27 | 0 | 0 | 0 | 0 | 0 | 0 |




(a) (b) (c)

Fig. 11: Cost $r(\tau)$ vs repair threshold $(\tau)$ for: (a) $\rho = 0.004$, (b) $\rho = 0.1$, (c) $\rho = 1$.

repair is optimal in terms of average repair cost. We investigated codes that perform repair through cooperation. We showed that similar to regenerating codes, one can derive optimal thresholds for cooperative regenerating codes. We also investigated the case when the repair process depends on the number of nodes under repair. Finally, we analyzed the case where we lift the restriction that once the repair process is initiated, no more departures occur. We showed that the initial fixed-rate repair model, which was simple enough to track analytically, is a good approximation of the complex model in the low $\rho$ regime, which is of interest. This allows us to use the optimal threshold results as well as other results we had from the simpler model.

As part of future work, we plan to consider a more advanced repair model in which fragment repairs occur under a fixed bandwidth constraint. This assumption makes the repair rate $\mu$ dependent on the repair threshold $\tau$. Furthermore, in this study, we did not take energy consumption into an account, i.e., due to encoding/decoding operations. A more advanced threshold optimization problem may include this additional cost, which can differ based on the coding schemes. Finally, generalizing the model to the case where repairs are initiated at every state with some probability and studying the cost vs. MTTDL tradeoff under this model is an interesting avenue for further research.

## REFERENCES

[1] R. Bhagwan, K. Tati, Y.-C. Cheng, S. Savage, and G. M. Voelker, "Total recall: system support for automated availability management." in *Proc. of the NSDI Conference*, 2004.

[2] F. Dabek, J. Li, E. Sit, J. Robertson, M. F. Kaashoek, and R. Morris, "Designing a DHT for low latency and high throughput," in *Proc. of the NDSI Symposium*, 2004.

[3] D. Borthakur, "HDFS architecture guide," http://hadoop.apache.org/common/docs/current/hdfs_design.pdf, 2008.

[4] Nasuni, "The state of cloud storage: 2015 industry report," http://www6.nasuni.com/rs/445-ZDB-645/images/Nasuni-White-Paper-2015-State-of-Cloud-Storage.pdf, 2015.

[5] M. Satyanarayanan, "The emergence of edge computing," *Computer*, vol. 50, no. 1, pp. 30–39, Jan 2017.

[6] N. Golrezaei, A. G. Dimakis, and A. F. Molisch, "Device-to-device collaboration through distributed storage," in *Proc. of the GLOBECOM Conference*, 2012.

[7] J. Pääkkönen, C. Hollanti, and O. Tirkkonen, "Device-to-device data storage for mobile cellular systems," in *Proc. of the GLOBECOM Workshops*, 2013.

[8] N. Golrezaei, A. G. Dimakis, and A. F. Molisch, "Scaling behavior for device-to-device communications with distributed caching," *IEEE Trans. on Information Theory*, vol. 60, no. 7, pp. 4286–4298, 2014.

[9] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 637–646, Oct 2016.

[10] V. Bioglio, F. Gabry, and I. Land, "Optimizing mds codes for caching at the edge," in *Proc. of the GLOBECOM Conference*, 2015.

[11] M. Xia, M. Saxena, M. Blaum, and D. Pease, "A tale of two erasure codes in hdfs." in *Proc. of the 13th USENIX Conference on File and Storage Technologies*, Santa Clara, CA, Feb. 2016.

[12] Y. Hu, Y. Xu, X. Wang, C. Zhan, and P. Li, "Cooperative recovery of distributed storage systems from multiple losses with network coding," *IEEE Journal on Selected Areas in Communications*, vol. 28, no. 2, pp. 268–276, 2010.

[13] H. Weatherspoon and J. D. Kubiatowics, "Erasure coding vs. replication: A quantitive comparison," in *Proc. of the 1st International Workshop on Peer-to-Peer Systems*, 2002.

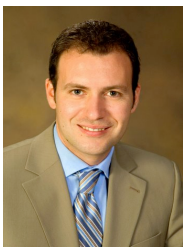
[14] F. Giroire, J. Monteiro, and S. Pérennes, "Peer-to-peer storage systems: a practical guideline to be lazy," in *Proc. of the GLOBECOM Conference*, 2010.

[15] A. G. Dimakis, P. B. Godfrey, Y. Wu, M. J. Wainwright, and K. Ramchandran, "Network coding for distributed storage systems," *IEEE Trans. on Information Theory*, vol. 56, no. 9, pp. 4539–4551, 2010.
[16] K. Rashmi, N. Shah, and P. Kumar, "Optimal exact-regenerating codes for distributed storage at the MSR and MBR points via a product-matrix construction," *IEEE Trans. on Information Theory*, vol. 57, no. 8, pp. 5227 –5239, Aug. 2011.
[17] I. Tamo, Z. Wang, and J. Bruck, "Zigzag codes: MDS array codes with optimal rebuilding," *IEEE Trans. on Information Theory*, vol. 59, no. 3, pp. 1597–1616, March 2013.
[18] V. Cadambe, S. Jafar, H. Maleki, K. Ramchandran, and C. Suh, "Asymptotic interference alignment for optimal repair of MDS codes in distributed storage," *IEEE Trans. on Information Theory*, vol. 59, no. 5, pp. 2974–2987, 2013.
[19] D. Ford, F. Labelle, F. I. Popovici, M. Stokely, V.-A. Truong, L. Barroso, C. Grimes, and S. Quinlan, "Availability in globally distributed storage systems," in *Proc. Ninth USENIX Symposium on Operating Systems Design and Implementation (OSDI'10)*, Vancouver, BC, CA, Oct. 2010.
[20] R. C. Singleton, "Maximum distance q -nary codes," *IEEE Trans. on Information Theory*, vol. 10, no. 2, pp. 116–118, 1964.
[21] M. Blaum, J. Brady, J. Bruck, and J. Menon, "EVENODD: An efficient scheme for tolerating double disk failures in RAID architectures," *IEEE Trans. on Computers*, vol. 44, no. 2, pp. 192–202, Feb. 1995.
[22] C. Huang and L. Xu, "STAR: An efficient coding scheme for correcting triple storage node failures," in *Proc. of the USENIX Conference on File and Storage Technologies*, 2005.
[23] B. Calder, Wang *et al.*, "Windows Azure Storage: A highly available cloud storage service with strong consistency," in *Proc. of the Twenty-Third ACM Symposium on Operating Systems Principles*, 2011.
[24] D. Ford, F. Labelle, F. I. Popovici, M. Stokely, V.-A. Truong, L. Barroso, C. Grimes, and S. Quinlan, "Availability in globally distributed storage systems." in *Proc. of the OSDI Conference*, 2010.
[25] A.-M. Kermarrec, N. Le Scouarnec, and G. Straub, "Repairing multiple failures with coordinated and adaptive regenerating codes," in *Proc. of the International Symposium on Network Coding (NetCod)*, 2011.
[26] K. W. Shum and Y. Hu, "Cooperative regenerating codes," *IEEE Trans. on Information Theory*, vol. 59, no. 11, pp. 7229–7258, Nov 2013.
[27] X. Wang, Y. Xu, Y. Hu, and K. Ou, "Mfr: Multi-loss flexible recovery in distributed storage systems," in *Proc. of the International Conference on Communications (ICC)*, 2010.
[28] A. S. Rawat, O. O. Koyluoglu, and S. Vishwanath, "Centralized repair of multiple node failures with applications to communication efficient secret sharing," *arXiv preprint arXiv:1603.04822*, 2016.
[29] M. Burrows, "The chubby lock service for loosely-coupled distributed systems," in *Proc. Seventh USENIX Symposium on Operating Systems Design and Implementation (OSDI'06)*, Seattle, WA, Nov. 2006.
[30] P. Hunt, M. Konar, F. P. Junqueira, and B. Reed, "Zookeeper: Wait-free coordination for internet-scale systems." in *Proc. USENIX Annual Technical Conference*, Boston, MA, June 2010.
[31] J. Pääkkönen, P. Dharmawansa, C. Hollanti, and O. Tirkkonen, "Distributed storage for proximity based services," in *Proc. of the Swedish Communication Technologies Workshop (Swe-CTW)*, 2012.
[32] J. Pääkkönen, C. Hollanti, and O. Tirkkonen, "Device-to-device data storage with regenerating codes," in *Multiple Access Communications*. Springer, 2015, pp. 57–69.
[33] J. Pedersen, I. Andriyanova, F. Brännström *et al.*, "Distributed storage in mobile wireless networks with device-to-device communication," *arXiv preprint arXiv:1601.00397*, 2016.
[34] B. Eisenberg, "On the expectation of the maximum of iid geometric random variables," *Statistics & Probability Letters*, vol. 78, no. 2, pp. 135–143, 2008.

**Gokhan Calis** is Data Scientist at Western Digital. He received his B.S. and M.S. degrees in Electrical and Electronics Engineering from Bilkent University, Ankara, Turkey, in 2011 and 2013, respectively. He received his Ph.D. degree in Electrical and Computer Engineering from University of Arizona, Tucson, AZ, in 2017. His research interests include information theory, distributed storage networks and machine learning.

**Swetha Shivaramaiah** is a Software Engineer at Electronic Arts, where she is a part of Cloud Engineering team. She received B.E degree in electronics and communication engineering from Visvesvaraya Technological University, India and M.S degree in Electrical and Computer engineering from University of Arizona in 2012 and 2015 respectively. Her research work while at the University of Arizona focused on repair strategies for mobile distributed storage systems.

**O. Ozan Koyluoglu** received the B.S. degree in electrical and electronics engineering from Bilkent University, Ankara, Turkey, in 2005, and the M.S. and Ph.D. degrees in electrical and computer engineering from The Ohio State University, Columbus, OH in 2007 and in 2010, respectively. From October 2010 to January 2011, he was with the Wireless Communication Theory Research Group at Nokia Bell Labs. From January 2011 to August 2013, he was a postdoctoral fellow at The University of Texas at Austin. From August 2013 to January 2017, he was an assistant professor at the Department of Electrical and Computer Engineering, The University of Arizona. Since January 2017, he has been affiliated with University of California, Berkeley. His current research interests are in the areas of information theory, machine learning, distributed storage/computing, networks, and computational neuroscience.



**Loukas Lazos** is an associate professor of Electrical and Computer Engineering at the University of Arizona. Dr. Lazos received his Ph.D. degree in Electrical Engineering from the University of Washington in 2006. His research interests are in the areas of security and privacy, networking, and wireless communications. His current research focuses on detection, mitigation, and visualization of security threats; secure device bootstrapping and key management for wireless networks; secure channel access protocol design for emerging wireless technologies; secure and fair resource allocation for heterogeneous coexisting systems and privacy in wireless communications. Dr. Lazos is the recipient of the US National Science Foundation (NSF) Faculty Early CAREER Development Award (2009) for his research in security of multi-channel wireless networks. He has served as a technical program chair for the IEEE CNS conference, the IEEE GLOBECOM symposium on communications and information systems security and the IEEE DSPAN workshop. He is an associate editor for the IEEE Transactions on Information and Forensics Security journal and the IEEE Transactions on Mobile Computing journal. He has also served and continues to serve on the organization and technical program committees of many international conferences and on expert panels of several government agencies.

# Repair Strategies for Mobile Storage Systems

Gokhan Calis, *Member, IEEE*, Swetha Shivaramaiah, O. Ozan Koyluoglu, *Member, IEEE*, and Loukas Lazos, *Member, IEEE*

## APPENDIX A
## PROOF OF PROPOSITION 1

*Proof.* To determine $\tau^*$, we compare $r_D(d)$ with the average repair cost at all other possible regeneration states $d+\delta$, for $1 \leq \delta \leq n-d-1$, and check if

$$r_D(d) \leq r_D(d+\delta), \ \ \forall \delta \in [1, n-d-1]. \tag{50}$$

This method is preferred because a straightforward minimization of $r_D(\tau)$ through differentiation is involved due to the harmonic sums. Substituting $r(\tau)$ from (7) to (50) yields,

$$\frac{\gamma(n-d)\lambda\mu}{\mu H_{n,d}+\lambda} \leq \frac{\gamma(n-d-\delta)\lambda\mu}{\mu H_{n,d+\delta}+\lambda}, \tag{51}$$

$$\rho \leq \frac{(n-d)H_{d+\delta,d}}{\delta} - H_{n,d}. \tag{52}$$

The inequality in (52) yields the maximum $\rho$ for which $r_D(\tau)$ is minimized at state $\tau = d$. We now examine the behavior of the right hand side (RHS) in (52) as a function of $\delta$ for fixed $n$ and $d$. The RHS in (52) has the same monotonicity as the function $f(d,\delta) = \frac{H_{d+\delta,d}}{\delta}$. In Lemma 3, we show that $f$ is monotonically decreasing with $\delta$. As a result, the departure-to-repair rates $\rho$ for which (50) holds are also monotonically decreasing with $\delta$. Substituting the maximum $\delta$ (i.e., $\delta = n-d-1$) to the RHS in (52) yields a departure rate bound

$$\rho \leq \frac{H_{n-1,d}}{n-d-1} - \frac{1}{n}, \tag{53}$$

for which $r_D(d) \leq r_D(d+\delta), \forall \delta \in [1, n-d-1]$. In this case, minimization of $r_D(\tau)$ is achieved at $\tau^* = d$.

We now prove that for rates $\rho > \frac{H_{n-1,d}}{n-d-1} - \frac{1}{n}$, the average cost $r_D(\tau)$ is minimized when $\tau = n-1$. Following a similar reasoning, we compare $r_D(\tau)$ at $\tau = n-1$ with $r_D(\tau)$ at any other possible regeneration threshold. We consider $r_D(n-1) \leq r(n-\delta-1)$, where $1 \leq \delta \leq n-d-1$. By substituting $r_D(\tau)$ from (7) and simplifying, it follows that

$$\rho \geq \frac{H_{n-1,n-\delta-1}}{\delta} - \frac{1}{n}. \tag{54}$$

The RHS of (54) has the same monotonicity as the function $g(n-1,\delta) = \frac{H_{n-1,n-1-\delta}}{\delta}$. In Lemma 4, we show that $g$ is monotonically increasing with $\delta$. Therefore, the minimum $\rho$ for which $r_D(n-1) \leq r_D(n-1-\delta), \ \forall \delta \in [1, n-d-1]$ is obtained when $\delta = n-d-1$. Substituting this $\delta$ to (54) completes the proof. □

**Lemma 3.** *The function*

$$f(x,\delta) = \frac{H_{x+\delta,x}}{\delta} \tag{55}$$

*is a monotonically decreasing function over integers $\delta > 0$ for any given integer $x > 0$.*

*Proof.* We will show that $f(x,\delta+1) < f(x,\delta)$ for any integer $\delta > 0$, which implies the monotonically decreasing assertion in the lemma. We have

$$f(x,\delta) - f(x,\delta+1) = \frac{1}{\delta}\left(\sum_{i=x+1}^{x+\delta}\frac{1}{i}\right) - \frac{1}{\delta+1}\left(\sum_{i=x+1}^{x+\delta+1}\frac{1}{i}\right)$$
$$\overset{(a)}{=} \frac{S}{\delta} - \frac{S+\frac{1}{x+\delta+1}}{\delta+1} = \frac{1}{\delta+1}\left(S\left(\frac{\delta+1}{\delta}-1\right) - \frac{1}{x+\delta+1}\right)$$
$$= \frac{1}{\delta(\delta+1)}\left(S - \frac{\delta}{x+\delta+1}\right) \overset{(b)}{>} 0,$$

where in (a) we define the sum $S = \sum_{i=x+1}^{x+\delta}\frac{1}{i}$, and (b) follows as there are $\delta$ terms in $S$ and each term is strictly greater than $\frac{1}{x+\delta+1}$. □

**Lemma 4.** *The function*

$$g(x,\delta) = \frac{H_{x,x-\delta}}{\delta} \tag{56}$$

*is a monotonically increasing function over integers $\delta \in [0, x-1]$ for any given integer $x > 0$.*

*Proof.* We will show that $g(x,\delta+1) > g(x,\delta)$ for any integer $\delta > 0$, which implies the monotonically increasing assertion in the lemma. We have

$$g(x,\delta+1) - g(x,\delta) = \frac{1}{\delta+1}\left(\sum_{i=x-\delta}^{x}\frac{1}{i}\right) - \frac{1}{\delta}\left(\sum_{i=x-\delta+1}^{x}\frac{1}{i}\right)$$
$$\overset{(a)}{=} \frac{S+\frac{1}{x-\delta}}{\delta+1} - \frac{S}{\delta} = \frac{1}{\delta+1}\left(S\left(1-\frac{\delta+1}{\delta}\right) + \frac{1}{x-\delta}\right)$$
$$= \frac{1}{\delta(\delta+1)}\left(\frac{\delta}{x-\delta} - S\right) \overset{(b)}{>} 0,$$

where in (a) we define the sum $S = \sum_{i=x-\delta+1}^{x}\frac{1}{i}$, and (b) follows as there are $\delta$ terms in $S$ and each term is strictly smaller than $\frac{1}{x-\delta}$. □

## APPENDIX B
## PROOF OF LEMMA 1

In Proposition 1, we determined the $\rho$ regime for which lazy repair is more efficient than eager repair, given fixed code parameters. To prove Lemma 1, it suffices to show that the highest rate $\rho = \frac{H_{n-1,d}}{n-d-1} - \frac{1}{n}$ for which $\tau^* = d$ is strictly positive for any $n$ and $d$ (recall that by definition, $n > d$).

We prove this fact by employing a lower bound on $\frac{H_{n-1,d}}{n-d-1}$ , which is proved in Lemma 5.

$$\frac{H_{n-1,d}}{n-d-1} \geq \frac{1}{n-1} \overset{(a)}{\Rightarrow} \tag{57}$$

$$\frac{H_{n-1,d}}{n-d-1} > \frac{1}{n} \tag{58}$$

$$\frac{H_{n-1,d}}{n-d-1} - \frac{1}{n} > 0. \tag{59}$$

where in (a), we substituted $\frac{1}{n-1}$ with the strictly smaller term $\frac{1}{n}$.

**Lemma 5.** *The function*

$$f(x,\delta) = \frac{H_{x+\delta,x}}{\delta} \tag{60}$$

*is bounded by*

$$\frac{1}{x+\delta} \leq \frac{H_{x+\delta,x}}{\delta} < \frac{1}{\delta}, \tag{61}$$

*for all positive integers $x$ and $\delta$.*

*Proof.* First, we show that $\frac{H_{x+\delta,x}}{\delta} \geq \frac{1}{x+\delta}$, for all $x > 0$ and $\delta > 0$.

$$\frac{H_{x+\delta,x}}{\delta} = \frac{\sum_{i=1}^{x+\delta} \frac{1}{i} - \sum_{i=1}^{x} \frac{1}{i}}{\delta} = \frac{\frac{1}{x+1} + \frac{1}{x+2} + \ldots + \frac{1}{x+\delta}}{\delta}$$
$$\overset{(a)}{>} \frac{\frac{1}{x+\delta} + \frac{1}{x+\delta} + \ldots + \frac{1}{x+\delta}}{\delta} = \frac{\delta \frac{1}{x+\delta}}{\delta} = \frac{1}{x+\delta},$$

where (a) follows by substituting the first $\delta - 1$ terms in the nominator with strictly smaller terms. The equality in this lower bound holds when $\delta = 1$. We now show the upper bound.

$$\frac{H_{x+\delta,x}}{\delta} = \frac{\sum_{i=1}^{x+\delta} \frac{1}{i} - \sum_{i=1}^{x} \frac{1}{i}}{\delta} = \frac{\frac{1}{x+1} + \frac{1}{x+2} + \ldots + \frac{1}{x+\delta}}{\delta}$$
$$\overset{(b)}{<} \frac{\frac{1}{x} + \frac{1}{x} + \ldots + \frac{1}{x}}{\delta} = \frac{\delta \frac{1}{x}}{\delta} = \frac{1}{x},$$

where (b) follows by substituting the $\delta$ terms in the nominator with strictly larger terms. □

## APPENDIX C
## PROOF OF PROPOSITION 2

*Proof.* The proof follows along the same lines as Proposition 1. We compare the repair cost at $r(d)$ with the repair cost at any other possible state $d - \delta$ for $1 \leq \delta \leq d - k$ to check when the inequality

$$r(d) \leq r(d-\delta), \quad \delta \in [1, d-k] \tag{62}$$

is satisfied. Substituting for $r(\tau)$ using (7), we obtain

$$\frac{\lambda\mu(n-d)\gamma}{\mu H_{n,d} + \lambda} \leq \frac{\lambda\mu(k\alpha\delta + (n-d)\gamma)}{\mu H_{n,d-\delta} + \lambda}, \tag{63}$$

from which we get:

$$\rho \geq \frac{(n-d)\gamma H_{d,d-\delta}}{k\alpha\delta} - \frac{1}{n} - H_{n-1,d} \tag{64}$$

Expression (64) yields a bound on the minimum $\rho$ for which the optimal repair threshold is $\tau^* = d$. We notice that RHS of the inequality above has the same monotonicity as the function $g(d,\delta) = \frac{H_{d,d-\delta}}{\delta}$ defined in Lemma 4 in Appendix A, from which we observe that this function is a monotonically increasing function of $\delta$. Substituting the maximum $\delta^* = d - k$ yields the departure-to-repair rate bound,

$$\rho \geq \frac{\gamma(n-d)H_{d,k}}{k\alpha(d-k)} - H_{n,d}, \tag{65}$$

for which $r(d) \leq r(d-\delta), \forall \delta \in [1, d-k]$. For this rate regime, the optimal repair threshold is at $\tau^* = d$.

We now prove that for rates $\rho \leq \frac{\gamma(n-d)H_{d,k}}{k\alpha(d-k)} - H_{n-1,d} - \frac{1}{n}$, the average cost $r(\tau)$ per unit of time is minimized when $\tau = k$. We compare $r(k)$ with $r(k+\delta)$ to analyze when the following inequality holds.

$$r(k) \leq r(k+\delta), \quad \delta \in [1, d-k]. \tag{66}$$

On substituting for $r(\tau)$ from (7), we get:

$$\frac{\lambda\mu(k\alpha(d-k) + \gamma(n-d))}{\mu H_{n,k} + \lambda} \leq \frac{\lambda\mu(k\alpha(d-k-\delta) + \gamma(n-d))}{\mu H_{n,k+\delta} + \lambda},$$

from which we obtain

$$\rho \leq \frac{(k\alpha(d-k) + \gamma(n-d))H_{k+\delta,k}}{k\alpha\delta} - \frac{1}{n} - H_{n-1,k}. \tag{67}$$

The expression above yields a bound on the maximum departure-to-repair rate $\rho$ for which the optimal repair threshold is $\tau^* = k$. We now study the behavior of (67) as a function of $\delta$ for fixed $n, k$ and $d$. We notice that RHS of the inequality above has the same monotonicity as the function $f(k,\delta) = \frac{H_{k+\delta,k}}{\delta}$ defined in Lemma 3 in Appendix A, from which we observe that this function is a monotonically decreasing function of $\delta$. Therefore, $\delta^* = d - k$ yields the minimum value for the RHS of (67), which implies that when

$$\begin{aligned} \rho &\leq \frac{(k\alpha(d-k) + \gamma(n-d))H_{d,k}}{k\alpha(d-k)} - H_{n-1,k} - \frac{1}{n} \\ &= \frac{\gamma(n-d)H_{d,k}}{k\alpha(d-k)} - H_{n,d}, \end{aligned} \tag{68}$$

$r(\tau)$ is optimized at $\tau^* = k$. □

## APPENDIX D
## PROOF OF LEMMA 2

*Proof.* We prove Lemma 2 by finding the relationship between $n, k, \gamma$, and $\alpha$ for which the upper bound on the rate $\rho$ when $\tau^* = k$ becomes negative.

$$\begin{aligned} \frac{\gamma(n-d)H_{d,k}}{k\alpha(d-k)} - H_{n,d} &= (n-d)\left(\frac{\gamma H_{d,k}}{k\alpha(d-k)} - \frac{H_{n,d}}{n-d}\right) \\ &\overset{(a)}{<} (n-d)\left(\frac{\gamma}{k^2\alpha} - \frac{H_{n,d}}{n-d}\right) \\ &\overset{(b)}{\leq} (n-d)\left(\frac{\gamma}{k^2\alpha} - \frac{1}{n}\right) \\ &= \frac{n-d}{k^2\alpha}\left(n\gamma - k^2\alpha\right). \end{aligned}$$

where in (a) we have used Lemma 5 of Appendix B to substitute term $\frac{H_{d,k}}{d-k}$ with its upper bound $\frac{1}{k}$ and in (b), we have used the same lemma to substitute $\frac{H_{n,d}}{n-d}$ with its lower bound $\frac{1}{n}$. Setting $\frac{n-d}{k^2\alpha}\left(n\gamma - k^2\alpha\right) < 0$ yields $n\gamma < k^2\alpha$ since the multiplicative term is strictly positive. □

## APPENDIX E
## PROOF OF PROPOSITION 3

*Proof.* To determine $\tau^*$, we compare $r(k)$ with other possible repair states $k+\delta$, i.e., we analyze when

$$r(k) \leq r(k+\delta), \tag{69}$$

for $1 \leq \delta \leq n-k-1$. On substituting for $r(\tau)$ from (11), we obtain

$$\frac{\lambda\mu\alpha(k+n-k-1)}{\mu H_{n,k}+\lambda} \leq \frac{\lambda\mu\alpha(k+n-k-\delta-1)}{\mu H_{n,k+\delta}+\lambda}.$$

We have

$$\rho \leq \frac{(k\alpha+\alpha(n-k-1))H_{k+\delta,k}}{\alpha\delta} - \frac{1}{n} - H_{n-1,k}. \tag{70}$$

Inequality (70) yields the maximum departure-to-repair rate $\rho$ for which it is more cost-efficient to repair at state $\tau = k$ than any other state $\tau = k+\delta$. We now examine the behavior of the RHS of (70) as a function of $\delta$ for fixed $k$ and $d$. We notice that RHS of the inequality above has the same monotonicity as the function $f(k,\delta) = \frac{H_{k+\delta,k}}{\delta}$ defined in Lemma 3 in Appendix A, from which we observe that this function is a monotonically decreasing function of $\delta$. Substituting $\delta^* = n-k-1$ to the RHS of (70) yields

$$\rho \leq \frac{kH_{n-1,k}}{(n-k-1)} - \frac{1}{n}, \tag{71}$$

for which the optimal repair threshold is at $\tau^* = k$.

We now evaluate if there is a departure rate regime for which the average cost per unit of time is minimized at $\tau^* = n-1$. That is, we analyze when

$$r(n-1) \leq r(n-\delta-1). \tag{72}$$

where $1 \leq \delta \leq n-k-1$. On substituting for $r(\tau)$ from (11), we get

$$\frac{\lambda\mu\alpha k}{\mu H_{n,n-1}+\lambda} \leq \frac{\lambda\mu\alpha(k+\delta)}{\mu H_{n,n-\delta-1}+\lambda} \Rightarrow \tag{73}$$

$$\rho \geq \frac{kH_{n-1,n-\delta-1}}{\delta} - \frac{1}{n}. \tag{74}$$

We notice that RHS of the inequality above has the same monotonicity as the function $g(n-1,\delta) = \frac{H_{n-1,n-1-\delta}}{\delta}$ defined in Lemma 4 in Appendix A, from which we observe that this function is a monotonically increasing function of $\delta$. Substituting $\delta^* = n-k-1$ yields

$$\rho \geq \frac{kH_{n-1,k}}{(n-k-1)} - \frac{1}{n}, \tag{75}$$

for which the optimal repair threshold is at $\tau^* = n-1$.

□

## APPENDIX F
## PROOF OF PROPOSITION 4

*Proof.* Let us consider the following inequality:

$$r_D(\tau) < r_C(\tau). \tag{76}$$

According to (11), the average repair cost $r_C(\tau)$ of centralized repair depends only on $\alpha$, when $n$, $k$, and $d$ are fixed. As $\alpha_{MSR} \leq \alpha_{MBR}$, MSR codes minimize $r_C(\tau)$. Thus, we select MSR codes for centralized repair in our comparison. Similarly, for given $n, k$, and $d$, the average repair cost $r_D(\tau)$ of distributed repair depends only on the repair bandwidth $\gamma$. As $\gamma_{MBR} \leq \gamma_{MSR}$, MBR codes are selected to minimize $r_D(\tau)$. Substituting (1) and (2) in $r_C(\tau)$ and $r_D(\tau)$, respectively, we obtain

$$\frac{\mathcal{M}(2d)(n-\tau)\lambda\mu}{k(2d-k+1)(\mu H_{n,\tau}+\lambda)} < \frac{\mathcal{M}(k+n-\tau-1)\lambda\mu}{k(\mu H_{n,\tau}+\lambda)}, \tag{77}$$

which implies

$$k+n-\tau-1 < 2d. \tag{78}$$

Inequality (78) determines the minimum number of surviving nodes for which MBR distributed repair emerges as the most cost-efficient strategy. The left hand side (LHS) of (78) is a decreasing function of $\tau$. Maximizing the LHS yields the relationship between $n, k$, and $d$ for which distributed MBR *always* outperforms centralized MSR. This occurs when $\tau = d$. Substituting $\tau = d$ results in $d > \frac{n+k-1}{3}$. If we reverse the direction of the inequality in (76), we obtain

$$2d < n+k-\tau-1. \tag{79}$$

Minimizing the RHS of (79) yields the relationship between $n, k$, and $d$ for which centralized MSR *always* outperforms distributed MBR. This occurs when $\tau = n-1$. Substituting $\tau = n-1$ results in $d < \frac{k}{2}$. However, by the definition of regenerating codes, we have $d \geq k$. Therefore, there is no condition for which centralized MSR repair always outperforms distributed MBR repair. □

## APPENDIX G
## PROOF OF PROPOSITION 5

*Proof.* To determine the optimal repair strategy we compare $r_C(\tau)$ with $r_D(\tau)$ for $k \leq \tau^* < d$

$$r_C(\tau) < r_D(\tau).$$

Substituting $r(\tau)$ for distributed repair and centralized repair from (7) and (11), respectively, we obtain:

$$(\alpha(k+n-\tau-1))\frac{\lambda\mu}{H_{n,\tau}+\lambda} < (\alpha k(d-\tau)+\gamma(n-d))\frac{\lambda\mu}{H_{n,\tau}+\lambda}. \tag{80}$$

For MSR codes, $\alpha_{MSR} \leq \gamma_{MSR}$ and for MBR codes, $\alpha_{MBR} = \gamma_{MBR}$. Thus, for each case, we have $\alpha \leq \gamma$. By choosing the lowest $\gamma$, we consider when the parameters satisfy

$$\begin{aligned} \alpha(k+n-\tau-1) &< \alpha k(d-\tau)+\alpha(n-d) \\ \Rightarrow k-1 &< k(d-\tau)-(d-\tau) \\ \Rightarrow k-1 &< (k-1)(d-\tau) \\ \Rightarrow \tau &< d. \end{aligned} \tag{81}$$

As $k \leq \tau^* < d$, inequality (81) is always true and hence, centralized repair outperforms distributed repair. As explained in Proposition 4, for centralized repair, MSR codes minimize the average repair cost rate per unit of time as compared to MBR codes. Thus, centralized repair using MSR codes yields the optimal repair strategy. □

## APPENDIX H
## PROOF OF PROPOSITION 6

*Proof.* Starting from state $n$, expected time to reach state $\tau$ is $\sum_{i=\tau+1}^{n} = \frac{1}{i\lambda} = \frac{H_{n,\tau}}{\lambda}$. Once the system is at state $\tau$, with probability $p = \frac{\tau\lambda}{\tau\lambda+\mu}$, the system will transition to state $\tau - 1$. If this occurs, the recovery will not be possible since the number of fragments are below the repair threshold. However, DSS can still serve the users if $\tau - 1 \geq k$. Until we reach state $k-1$, the data is not lost. The expected time to reach state $k-1$ from state $\tau$ is $\sum_{i=k}^{\tau} \frac{1}{i\lambda} = \frac{H_{\tau,k-1}}{\lambda}$. On the other hand, with probability $1-p$, recovery will be initiated and the system will be back to state $n$, which takes $\frac{1}{\mu}$ time. At that point, we will go thorough the same process again. Accordingly, we have

$$TDL = \begin{cases} \frac{H_{n,\tau}}{\lambda} + \frac{H_{\tau,k-1}}{\lambda}, & \text{w.p } p \\ \frac{2H_{n,\tau}}{\lambda} + \frac{1}{\mu} + \frac{H_{\tau,k-1}}{\lambda} & \text{w.p. } (1-p)p \\ \frac{3H_{n,\tau}}{\lambda} + \frac{2}{\mu} + \frac{H_{\tau,k-1}}{\lambda} & \text{w.p. } (1-p)^2 p \\ \dots & \dots \\ \frac{iH_{n,\tau}}{\lambda} + \frac{i-1}{\mu} + \frac{H_{\tau,k-1}}{\lambda} & \text{w.p. } (1-p)^{i-1} p \end{cases} \tag{82}$$

from which the expected time to data loss, $E[\text{TDL}]$, can be calculated. □

## APPENDIX I
## PROOF OF PROPOSITION7

*Proof.* Since in both scenarios, repairs are performed after $n - \tau$ node departures and node repairs are performed parallel, it's enough to compare only the required bandwidths. First, assume MSCR case, then we want to show that

$$\frac{\mathcal{M}(d+t_1-1)(n-\tau)}{k(d-k+t_1)} > \frac{\mathcal{M}(d+t_2-1)(n-\tau)}{k(d-k+t_2)}, \tag{83}$$

which is equivalent to

$$(t_2 - t_1)(k-1) > 0. \tag{84}$$

Since $t_2 > t_1$ and we can conclude that $c_{t_1}(\tau) > c_{t_2}(\tau)$ therefore $r_{t_1}(\tau) > r_{t_2}(\tau)$. Similarly, for MBCR case, we need to have

$$\frac{\mathcal{M}(2d+t_1-1)(n-\tau)}{k(2d-k+t_1)} > \frac{\mathcal{M}(2d+t_2-1)(n-\tau)}{k(2d-k+t_2)}, \tag{85}$$

from which we can obtain the same condition as (84). Henceforth, $c_{t_1}(\tau) > c_{t_2}(\tau)$ and $r_{t_1}(\tau) > r_{t_2}(\tau)$. Combining MSCR and MBCR cases, we can conclude that $r_{t_1}(\tau) > r_{t_2}(\tau)$. □