THRESHOLD BASED REPAIR STRATEGIES FOR MOBILE DISTRIBUTED STORAGE SYSTEMS

by

Swetha Shivaramaiah

A Thesis Submitted to the Faculty of the

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

In Partial Fulfillment of the Requirements
For the Degree of

MASTER OF SCIENCE

In the Graduate College

THE UNIVERSITY OF ARIZONA

2 0 1 5

STATEMENT BY AUTHOR

This thesis has been submitted in partial fulfillment of requirements for an advanced degree at The University of Arizona and is deposited in the University Library to be made available to borrowers under the rules of the Library.

Brief quotations from this thesis are allowable without special permission, provided that accurate acknowledgment of the source is made. Requests for permission for extended quotation from or reproduction of this manuscript in whole or in part may be granted by the head of the major department or the Dean of the Graduate College when in his or her judgement the proposed use of the material is in the interests of scholarship. In all other instances, however, permission must be obtained from the author.

SIGNED: _____

APPROVAL BY THESIS DIRECTOR

This thesis has been approved on the date shown below:

_____ _____

Loukas Lazos
Associate Professor of
Electrical and Computer Engineering

Date

ACKNOWLEDGMENTS

I would like to take this opportunity to thank all the wonderful people who made this thesis possible.

I would like to express my deepest gratitude to my academic advisor, Dr. Loukas Lazos, for his continuous support, excellent guidance and patience, throughout the course of my masters. I could not have imagined a better guide and advisor for my masters thesis. I am deeply indebted to Dr. Ozan Koyluoglu for his valuable suggestions and insightful advice on advancing my research. I would also like to thank Dr. Ivan Djordjevic for willing to be a part of my thesis defense committee.

I have been fortunate to have had amazing friends who have been there throughout to motivate and inspire me. I would especially like to thank my friends, Aakarsh Rao, Anurag Kumar, Ramya Malladi and Poonam Kadam for all the fun times and making my stay in Tucson a memorable one. I would also like to thank Monisha Shivanna for being a constant source of support.

To my lab-mates, thank you for the fun and support. You have been great sources of ideas and I am fortunate to have had such generous peers.

Last but not the least, I am deeply indebted to my family for their unconditional love and support that has inspired me to strongly pursue my goals. And, thank you God for everything.

I am forever indebted to all of you and many others who are not mentioned here.

To appa and amma,
whose unconditional love and encouragement
made this possible

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

More than 5 Exabytes of digital content is being created everyday. This content needs to be stored, indexed, cached, and searched by over 200 millions users on a daily basis. Moreover, this vast amount of data needs to become ubiquitously available across a variety of mobile and infrastructure-based platforms. The content explosion faced in modern times has placed an enormous strain on the existing Internet infrastructure. The Internet traffic has increased five-fold over the past five years and is expected to exceed 1,000 Exabytes per year by 2016. Over half of the Internet traffic is expected to be directed to non-PC devices (primarily mobile platforms) by the end of 2018. This mobile data explosion is primarily due to the rapid growth of smartphone devices.

One solution to the traffic explosion problem is to cache content as close as possible to the users that consume it. This strategy has been realized with the rapid deployment of content distribution networks (CDNs) over the past few years. CDNs alleviate the content distribution scalability problem. However, content is still primarily cached at the fixed infrastructure network. An alternative approach to temper network traffic is to exploit the extended storage capacity of modern mobile devices and cache content within a mobile storage cloud. The mobile devices could gain access to the content, without burdening the fixed infrastructure. However, storing data at a distributed mobile cloud raises challenging reliability problems. Device mobility could frequently render the cached content unavailable. To tackle this problem, reliable storage solutions for the mobile cloud become a necessity.

We study the data reliability problem for a community of devices forming a mobile cloud storage system. We consider the application of regenerating codes for maintaining a file within a geographically-limited area. 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 due to node

mobility. We show that at a low departure rate regime, a *lazy repair* strategy in which repairs are initiated after several nodes have left the system outperforms *eager repair* in which repairs are initiated after a single departure. This optimality is reversed when nodes are highly mobile. We further compare distributed and centralized repair strategies and derive the optimal repair threshold for minimizing the average repair cost per unit of time, as a function of underlying code parameters. Finally, we analyze storage reliability when repairs can be incomplete due to communication bandwidth constraints.

CHAPTER 1

INTRODUCTION

1.1 Motivation and Scope

Digital content is expected to be generated at a staggering rate of 40% in the next decade [12]. This content needs to be stored, indexed, cached, and searched by over 200 millions users on a daily basis. Also, this huge volume of data needs to be accessible across a variety of mobile and infrastructure-based platforms. Enormous strain has been levied on the infrastructure network because of this surge in Internet content. The Internet traffic has increased five-fold over the past five years and is expected to exceed 1,000 Exabytes per year by 2016 [6]. Over half of the Internet traffic is expected to be directed to non-PC devices (primarily mobile platforms) by the end of 2018 [6]. This mobile data explosion is primarily due to the rapid growth of smartphone devices.

One solution to the traffic explosion problem is to cache content as close as possible to the users that consume it. This strategy has been realized with the rapid deployment of content distribution networks (CDNs) over the past few years. CDNs alleviate the contend distribution scalability problem [4]. However, content is still primarily cached at the fixed infrastructure network. An alternative approach to temper network traffic is to exploit the extended storage capacity of modern mobile devices and locally cache content within a mobile storage cloud [17,26,29,31,33,34]. A mobile distributed storage system consists of a community of mobile devices that are capable of storing data. In such a storage scenario, a file $\mathcal{F}$ is stored within a geographically-limited area $\mathcal{A}$ by mobile devices located within $\mathcal{A}$. A user within $\mathcal{A}$ can download $\mathcal{F}$ from the community of mobile devices, without accessing the network infrastructure. This approach has the potential of reducing the bandwidth required to maintain the stored data, and easy the traffic on the infrastructure

network.

However, in a mobile storage system, devices move freely across the space. Mobility can lead to frequent data loss when devices depart from the area of interest $\mathcal{A}$. For all practical purposes, when a mobile device storing $\mathcal{F}$ or any fragment of $\mathcal{F}$ exits $\mathcal{A}$, the stored data is lost. To deal with such losses, redundancy is introduced in the form of data replication or coding [2, 38]. In replication, copies of $\mathcal{F}$ are stored at multiple devices within the community. Although replication serves the purpose of upholding the reliability of the system, it is expensive in terms of storage bandwidth as file copies must be stored at several nodes. To reduce storage overhead (as compared to replication schemes), more sophisticated coding schemes can be utilized. In most cases, this is achieved with erasure codes (see, e.g., [35]), where the file $\mathcal{F}$ is encoded into several fragments such that it can be reconstructed as long as a threshold number of fragments are available within $\mathcal{A}$. Various redundancy methods impose different storage overheads on the mobile devices to maintain a desired level of reliability, and erasure codes are known to be amongst the most storage-efficient methods to reliably maintain data [2, 46].

Despite the application of coding, a stored file $\mathcal{F}$ will eventually be lost when a threshold number of mobile devices (storage nodes) depart from $\mathcal{A}$. To maintain $\mathcal{F}$ over long periods of time, the mobile cloud system must be capable of repairing the lost data (that can correspond to file or redundancy fragments). A repair scenario is shown in Figure.1.1. A file $\mathcal{F}$ is broken to four fragments. One fragment is used for redundancy. The four fragments are then disseminated among nodes in $\mathcal{A}$. When a node departs, a fragment is lost and then repair process is initiated. During the repair process, the lost data is recovered by downloading fragments from the storage nodes that remain within $\mathcal{A}$. The amount of data downloaded for repair is referred to as *repair bandwidth*. For mobile communities, the repair bandwidth can be significant due to frequent fragment loss. Excessive file repairs can lead to rapid energy depletion and spectral inefficiencies. Thus, it is important to optimize the repair bandwidth of distributed storage systems.

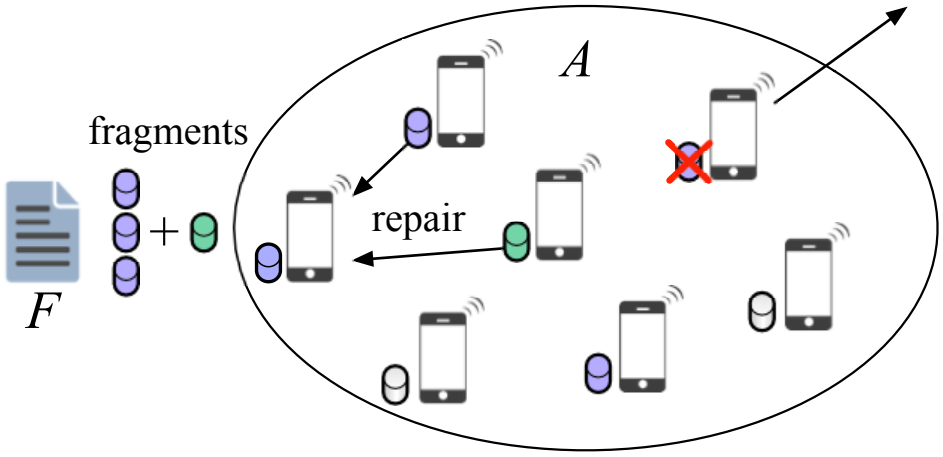

Figure 1.1: File maintenance through fragment repairs in a mobile cloud storage system.

The file repair problem for distributed storage systems has been primarily studied assuming that erasure codes are employed for redundancy [2, 16]. However, erasure codes are inefficient because the entire file $\mathcal{F}$ needs to be reconstructed to repair any lost fragment. The repair bandwidth can be reduced by employing regenerating codes, which allow the recovery of a lost fragment by downloading a smaller amount of data, without requiring the reconstruction of the file [9]. This process of obtaining only a lost fragment is termed as *regeneration*. Although regenerating codes lower the repair bandwidth, the design of an efficient repair strategy for the mobile cloud involves many parameters such as the redundancy factor of the code, the device departure rate from $\mathcal{A}$, the communication model for downloading data fragments, the threshold for starting maintenance operations, and the available communication bandwidth. In this thesis, *we jointly optimize the coding and file repair strategy for minimizing the cost of file maintenance in mobile cloud storage systems*. Specifically, we make the following contributions.

1.2 Main contributions and Thesis Organization

- We focus on threshold-based file maintenance strategies for mobile cloud storage systems. In such strategies, file repair is initiated when a threshold number of file fragments is lost. We analyze two repair strategies, namely distributed and centralized repair. In distributed repair, the new storage nodes directly download data from existing nodes to recover lost fragments. In centralized repair, a *leader* node first recovers the file $\mathcal{F}$ via reconstruction, then regenerates and transmits the remaining fragments to restore the system reliability. The two repair strategies are shown in Figures 1.2 and 1.3 . We derive the optimal repair threshold that minimizes the maintenance cost. We define the latter as long-run average repair cost per unit of time. We show that the optimal repair threshold depends on many system parameters and thus, provide departure rate based decision rule for choosing optimal repair strategy.

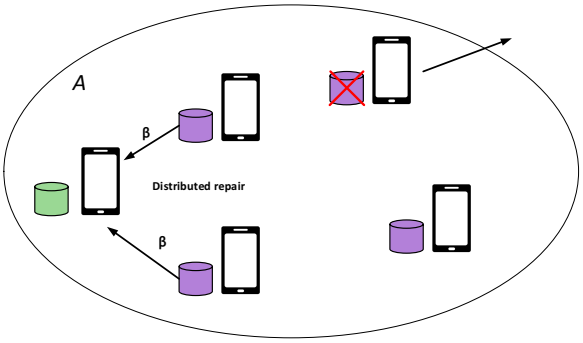

Figure 1.2: Distributed repair. A node directly downloads fragments from peers to repair lost fragments.

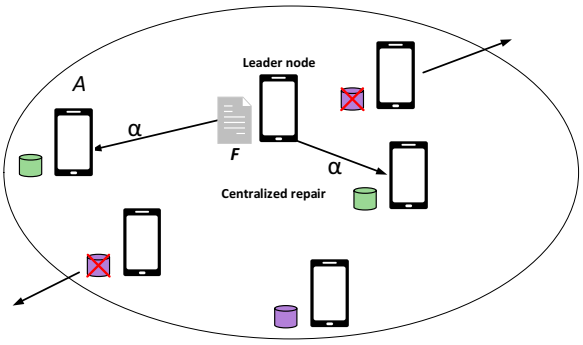

Figure 1.3: Centralized repair. A leader node distributes fragments to other nodes, after it reconstructs the file $\mathcal{F}$.

- Our results show that no one strategy is optimal for all possible system configurations. In high-mobility scenarios, regenerating after a single fragment loss minimizes the long-run average repair cost per unit of time. This repair policy is termed as eager repair [2]. In low-mobility scenarios, repairing after several fragments are lost yields a lower long-run average repair cost per unit of time. Delaying repair until several nodes have left $\mathcal{A}$ is termed as *lazy repair* [2]. For relatively static networks (very infrequent fragment loss), applying reconstruction and regeneration becomes the optimal strategy. We also determine the optimal repair strategy for a given departure rate.

- Finally, we analyze the storage system reliability when repairs can be incomplete due to communication bandwidth constraints. Specifically, we determine the probability of data loss and the mean time to data loss as a function of the repair threshold and other system parameters. We also discuss the tradeoff associated with system reliability metrics and average cost per time for various system parameters.

The remainder of the thesis is organized as follows. Chapter 2 highlights the related work. The model assumptions are presented in Chapter 3. We present the two repair strategies, namely distributed and centralized repair, in Chapter 4 and

5 respectively. In Chapter 6, we compare the repair policies and repair schemes under different code parameters. In Chapter 7, we perform analysis under the assumption of incomplete repairs and we summarize our conclusions in Chapter 8.

CHAPTER 2

BACKGROUND AND RELATED WORK

In this chapter, we present the basic concepts of reliable storage systems. We then extend our description to state-of-the-art in mobile storage.

2.1 Storage Reliability

Digital data is being created at a staggering rate and is expected to double every two years [12]. To accommodate storage demands, data centers are being deployed across the globe, which promise reliable data storage and fast retrieval. Reliability is ensured despite the fact that unreliable components are used for storage. This is achieved by deploying reliability algorithms that exploit data redundancy to maintain the stored content, despite the partial data loss. In redundant data storage, information is replicated or coded such that the original content can be recovered if some limited fraction of it is lost. The next subsection, we describe popular reliability methods for storage systems, namely *replication, erasure codes and regenerating codes.*

2.1.1 Replication

Replication is the most intuitive way to introduce redundancy. This method refers to the maintenance of verbatim copies of the same file $\mathcal{F}$ at multiple storage locations. It is currently employed in several storage systems including RAID systems. If n instances of a file $\mathcal{F}$ are available in the network, then up to $n-1$ simultaneous failures can be tolerated and this scheme is referred to as n-replication. Although replication is easy to implement and is adopted in numerous commercial platforms [5, 7, 15, 24, 25, 27, 40], it suffers from high storage overhead. The storage

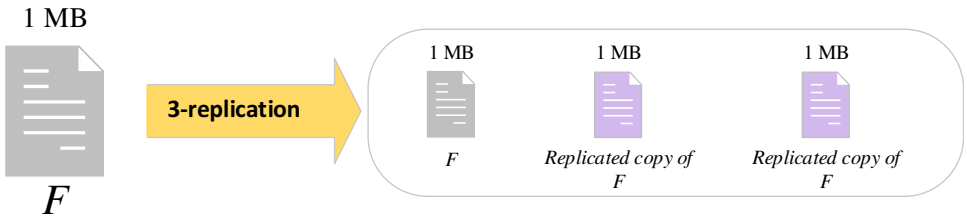

Figure 2.1: Storage of a 1MB file using 3-replication.

overhead grows linearly with the number of failures that must be tolerated. As an example, in Figure 2.1, the storage system must store 3 MB to maintain a 1MB file and sustain up to two node failures. The code rate for n-replication is equal to $\frac{1}{n}$. Here the code rate is defined as the ratio between the number of useful information bits stored over the total number of bits stored.

2.1.2 Erasure Codes

Erasure codes incur less storage overhead than replication to maintain the same reliability level. These codes were initially proposed to detect and correct errors that occur in the course of transmitting digital data. A class of erasure codes known as maximum distance separable codes(MDS) codes have optimal performance in terms of storage bandwidth. The basics of an MDS code (n, k) is explained with the help of an example as shown in Figure 2.2. A file $\mathcal{F}$ of size $\mathcal{M} = 1MB$ is first split into $k = 2$ fragments and then encoded into $n = 4$ fragments. Any subset of k out of n encoded fragments is sufficient to reconstruct $\mathcal{F}$. The n encoded fragments have the same length as the uncoded ones. Therefore, exactly $\mathcal{M}$ bytes are needed to reconstruct a file of $\mathcal{M}$ bytes. This corresponds to the same amount of data if replication were to be used. Reed-Solomon codes are a classical example of MDS codes and are already deployed in many existing storage systems (e.g. [3,11]). The code rate of an (n, k) erasure code is equal to $\frac{k}{n}$.

Although, erasure codes offer significant savings in terms of storage bandwidth,

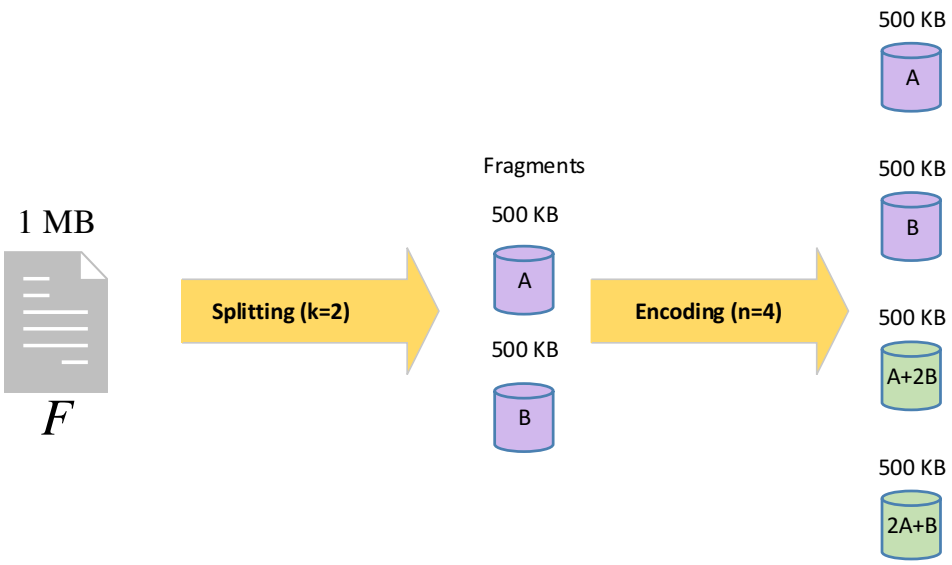

Figure 2.2: Storage of a 1MB file using (4,2) erasure codes.

the amount of data that needs to be retrieved for recovering lost data fragments can be prohibitive. This is because the entire file has to be reconstructed with every loss.

2.1.3 Regenerating Codes

An alternative to file reconstruction is regeneration. In regeneration, a lost encoded fragment can be repaired without recovering the entire file. For a fragment repair, it is sufficient to obtain the fragments of a subset of storage nodes. A file $\mathcal{F}$ of size $\mathcal{M}$ bits is stored in n storage nodes using a regenerating code with parameters $(n, k, d, \alpha, \gamma)$. Specifically, $\mathcal{F}$ is divided to k fragments, which are encoded to $n > k$ fragments such that any k encoded fragments can reconstruct $\mathcal{F}$. Each encoded fragment of size α symbols is stored in one of the n nodes. When a fragment is lost, a replacement node can regenerate the lost fragment by connecting to an arbitrary set of $d \geq k$ nodes out of the remaining $n - 1$ and downloading $\beta \leq \alpha$ symbols from each node. Therefore, the repair bandwidth for node regeneration is equal to $\gamma = d\beta$. Therefore, the repair bandwidth for node regeneration is equal

to $\gamma = d\beta$. The values for the fragment size (α) and the repair bandwidth (γ) can be calculated by using the following equations [9]:

$$\alpha(k,\gamma) = \begin{cases} \frac{\mathcal{M}}{k}, & \gamma \in [f(0), +\inf) \\ \frac{\mathcal{M}-g(i)\gamma}{k-1} & \gamma \in f(i), i = 1, \ldots, k-1 \end{cases}$$

$$f(i) = \frac{2\mathcal{M}d}{2k - i^2 - i + 2k + 2kd - 2k^2}$$

$$g(i) = \frac{(2d - 2k + i + 1)i}{2d}$$

The system designer can choose the parameter $i = 0, 1, \ldots k-1$ so that the resulting code meets the requirements of the application. The Minimum Storage Regenerating (MSR) code corresponds to i = 0. For the given distribution degree k, the MSR code has the smallest possible value of the fragment size (α). The MSR fragment size and the MSR repair bandwidth parameter $\beta = \frac{\gamma}{d}$ becomes

$$(\alpha_{MSR}, \gamma_{MSR}) = \left(\frac{\mathcal{M}}{k}, \frac{\mathcal{M}d}{k\,(d-k+1)}\right). \tag{2.1}$$

For MSR codes, $\alpha_{MSR} \leq \gamma_{MSR}$ and hence, per-node storage is smaller than the repair bandwidth. MBR codes, on the other hand, have minimum possible repair bandwidth (achieved when $\gamma = \alpha$), and operate at

$$(\alpha_{MBR}, \gamma_{MBR}) = \left(\frac{2\mathcal{M}d}{2kd - k^2 + k}, \frac{2\mathcal{M}d}{2kd - k^2 + k}\right). \tag{2.2}$$

The trade-off between storage bandwidth and repair bandwidth is as shown is in Figure 2.3. Figure 2.4 shows an example of $(n, k, d) = (4, 2, 3)$ regenerating codes. Here, the file $\mathcal{F}$ of size $\mathcal{M} = 1MB$ is first split into $k = 2$ fragments with each node storing $\alpha = 500KB$. Then, these fragments are encoded to obtain $n = 4$ fragments. A failed node in this scenario can regenerate by requesting fragments of size $\beta = 250KB$ from $d = 3$ surviving nodes. Thus, the repair bandwidth required to regenerate is given by $d\beta = 750KB$. On the other hand, the repair bandwidth required to reconstruct a file $\mathcal{F}$ is given by, $k\alpha = 1000KB$. Thus, the cost of regeneration is lesser than that of reconstruction.

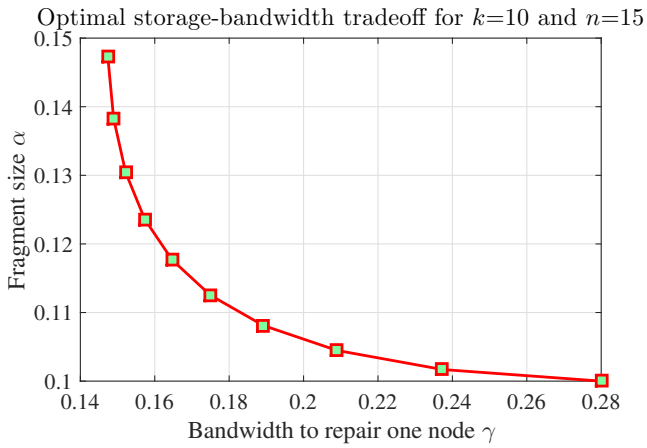

Figure 2.3: Tradeoff between repair bandwidth and storage bandwidth [9].

MSR and MBR codes achieve functional and exact repair. In functional repair, lost data fragments are replaced with functionally equivalent fragments, such that the desired degree of redundancy is maintained and hence, preserves only the recoverability property. In contrast to functional repair, exact repair is a stricter requirement in which corrupted data blocks are replaced with their exact replicas. Exact repair is preferred over functional repair in practical systems because the additional communication overheads involved during the repair is obliviated.

2.2 Related Work

The problem of reliable storage has been addressed in many different contexts. Peer-to-peer storage and mobile cloud storage systems are the two architectures most relevant to the setup considered in this thesis. In the following subsections, we describe the state-of-the-art for the two aforementioned architectures.

2.2.1 Peer-to-Peer Storage Systems

P2P systems consists of independent storage nodes that are distributed in a network. P2P systems exploit the locality feature and hence are considered to be

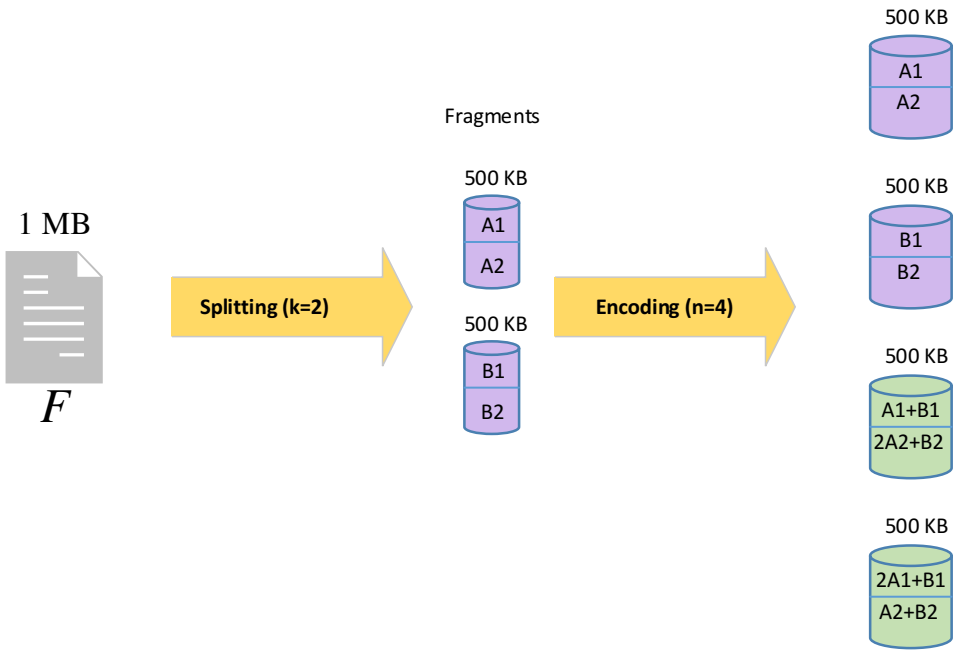

Figure 2.4: Storage of a 1MB file using a (4,2,3) regenerating code.

an interesting alternative to traditional centralized data centers. Large-scale P2P storage systems have been studied in different contexts [2, 8, 38, 40]. A number of works have investigated the prospect of employing replication against erasure codes for redundancy management [2, 5, 16, 20, 24, 39].

Utard and Vernois [44] consider a P2P storage system consisting of independent nodes which share their disk space for storing data fragments. The nodes are free to leave the system at anytime and thus, when a node leaves, the fragments stored at that node are lost. The authors investigate two redundancy mechanisms: replication and erasure codes. They determine the expected data lifetime for the two schemes using a Markov chain model, and show that it is better to use replication instead of erasure codes when the peer availability is very small. In contrast, Weatherspoon and Kubiatowicz [46] show that, for a system consisting of independently, identically distributed failing disks, erasure codes use an order of magnitude less repair bandwidth and storage space than replication to provide the same sys-

tem durability. This conclusion is based on quantitative analysis in terms of mean time to failure (MTTF).

Rodrigues and Liskov examine a cooperative data storage model [39]. They compare replication with a hybrid solution (mixing both replication and erasure codes). They studied real world traces (Overnet, Farsite and Planetlab) and proposed a failure model based on a membership timeout. Membership timeout is a metric that measures the system delay in responding to failures. After a timeout, the node state of a host that stores data changes to "failed", and their data need to be repaired. They state that erasure codes fare better compared to replication for scenarios of low server availability. In some cases, however, the complexity in terms of encoding and design of deploying erasure codes does not pay off the gains in storage efficiency.

Bhagwan et al. [2] proposed a P2P storage system called *TotalRecall* which is also based on erasure codes. They propose two dynamic repair strategies, namely eager repair and lazy repair. In eager repair, corrupted data blocks are immediately repaired upon detection. In lazy repair, data is recovered only after a threshold number of data blocks corrupted. The two strategies trade off reliability for network bandwidth efficiency. The lazy repair strategy incurs a lower network overhead for repairing corrupt data blocks at the expense of higher probability of data loss. The authors compare the repair bandwidth required by each policy using an empirical trace of P2P host availability. Our work analyzes similar threshold repair strategies for the mobile cloud environment. Additionally, we derive closed-form expressions for the repair bandwidth, but for regenerating codes, and optimally tune the repair threshold.

Giroire et al. [16] analytically evaluated network bandwidth metrics for the lazy repair strategy, using Markov models. Specifically, they computed the average required bandwidth per peer, the data loss rate and the peak of bandwidth consumption. In their analysis, a constant reconstruction time was assumed when performing repairs. Their work serves as a guideline for system designers to tune

system parameters depending on the desired level of reliability. They conclude that for a given reliability, lazy repair strategy achieves better utilization of bandwidth at the cost of storage bandwidth.

Works on P2P storage investigate the use of erasure codes and/or replication to maintain the redundancy of the system. In our work, we consider the use of regenerating codes [9] in mobile cloud storage systems. Previous works related to regenerating codes focus on the code construction [19, 23, 37, 41] and studying the properties of certain regenerating codes. For example, exact regenerating codes (see e.g. [18], [36], [45]) are codes that are able to reconstruct an exact copy of the lost data fragment. Deterministic code construction [47] allows for easily maintainable implementations. More recently, quasi-cyclic regenerating codes [13] have been introduced and shown to be efficient, simple regenerating codes. A few studies have been conducted on the practical implementation of regenerating codes: e.g. [10] concentrates on applying regenerating codes to peer-to-peer backup systems and in [21], the authors studied the impact of various parameters of regenerating codes at the system level rather than in terms of a single device. It also compares the computational costs of various implementations of regenerating codes and acts as a good guide in choosing the parameters in design of regenerating codes.

2.2.2 Mobile Cloud Storage Systems

Coded storage has also found application in wireless P2P storage systems for applications such as video sharing. The increased storage capacity of wireless devices has paved way for wireless P2P storage systems. While coding has been suggested to improve the performance of caching in terms of capacity and energy consumption [1, 22, 48], very few works offer solutions for keeping cached files available when the devices move out of the coverage area. Some works that consider the file maintenance problem are described below.

In [34], to increase the reliability of transmissions within the storage community, packet level erasure coding is investigated. A sparse delay tolerant network

consisting of mobile nodes that are capable of sending, forwarding, and receiving requests for resources is analyzed. The authors investigate the impact of different redundancy mechanisms on the performance of mobile nodes that obtain resources from an infrastructure network. They observe that erasure coding at the application layer may improve end-end delivery performance.

In [32], a wireless P2P storage system consisting of mobile users and a base-station is considered. It is assumed that the energy consumed in downloading a file requested by a user from the base station is greater than that consumed when data is transmitted between two mobile users. The mobile nodes cache data and upon request, can exchange data. They derive closed form expressions for expected total cost for two schemes; simple caching and redundant caching. Simple caching involves the file being stored on one of the local nodes in the network and hence, a new node obtains file contacting this node. In redundant caching, regenerating codes are used to cache the file on the storage nodes. On analyzing this system, it is proved that the expected total cost of 2-replication is lower than that of scheme with regenerating codes.

In [30], the authors assume a similar model as in [32]. They show that regenerating codes can be used to decrease the energy consumption of mobile cloud storage. They conclude that if the energy consumption per bit for the data transmission between two nodes is less expensive as compared to that between a node and a remote source, then regenerating codes decrease the overall energy consumption. The drawback of the analysis presented in [30] and [32] is that eager repair is considered. Also, their analysis assumes fixed code parameters that do not necessarily exploit the advantages provided by regeneration. The repair bandwidth associated with reconstruction is equivalent to that associated with regeneration in this case.

As an extension of [30, 31] and assuming the same model, in [33], the authors address the problem of tolerating multiple simultaneous failures. They investigate the performance of regenerating codes in terms of the total energy consumption of a cellular network. They show that large performance gains can be obtained by

employing regenerating codes. Also, they show that the popularity of a file has an effect on the gains associated with using redundancy. They also provide decision rules for choosing between simple caching, replication, MSR and MBR codes. These rules are based on numerical results on certain application scenarios. In this thesis, we analytically provide decision rules to choose optimal repair strategies apart from choosing optimal codes, MBR or MSR codes, that minimizes repair cost.

CHAPTER 3

SYSTEM MODEL ASSUMPTIONS

In this chapter, we describe the system model assumptions. The notation adopted in the rest of the thesis is presented in Table 3.1.

Table 3.1: Summary of Notations

Parameter	Description
$\mathcal{F}$	file
$\mathcal{M}$	file size in bits
$\mathcal{A}$	geographically-limited area where $\mathcal{F}$ is maintained
k	number of fragments of before file encoding
n	number of fragments of after file encoding
d	minimum number of fragments required for regeneration
α	fragment size in bits
β	repair fragment size for regeneration
γ	repair bandwidth
λ	node departure rate from $\mathcal{A}$
μ	repair rate
τ	repair threshold
B	communication bandwidth
$c(\tau)$	repair cost
$r(\tau)$	long-run average repair cost per unit time
$MTTDL$	mean time to data loss
P_{DL}	probability of data loss

3.1 Network model

We consider the mobile cloud storage system show in Figure 3.1. The system consists of mobile storage nodes that enter. When a node departs from $\mathcal{A}$, its data is lost. As we are interested in the system performance due to network dynamics, we do not consider data loss due to hardware failures. This is a reasonable assumption for the mobile environment in which data loss due to node departure occurs orders of magnitude more frequently than hardware failure.

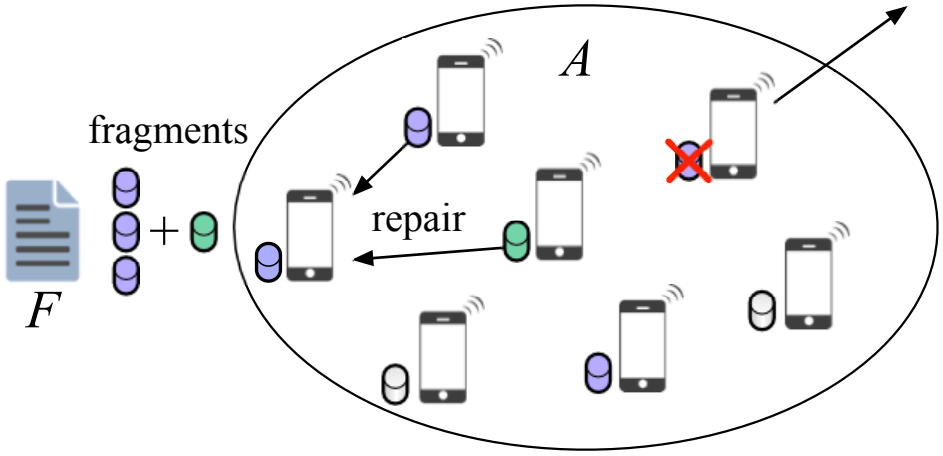

Figure 3.1: File maintenance through fragment repairs in a mobile cloud storage system.

Following the model of prior works [16, 32], the time that each storage node resides within $\mathcal{A}$ is an exponentially distributed random variable with parameter λ. The resident times of various storage nodes within $\mathcal{A}$ are assumed to be independent. Nodes in $\mathcal{A}$ are assumed to form a one-hop broadcast network. That is, transmissions from one node are received by all nodes within $\mathcal{A}$. This model also represents communications between nodes in multihop topologies at the logical level (by abstracting the broadcast relay operation).

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 $(n, k, d, \alpha, \gamma)$. Specifically, $\mathcal{F}$ is divided to k fragments, which are encoded to $n > k$ fragments such that any k encoded fragments can reconstruct $\mathcal{F}$. Each encoded fragment of size α symbols is stored in one of the n mobile nodes within $\mathcal{A}$. In case of a node departure from $\mathcal{A}$, a replacement node can regenerate the lost fragment by connecting to an arbitrary set of $d \geq k$ nodes out of the remaining $n-1$ and downloading $\beta \leq \alpha$ symbols from each node. Therefore, the repair bandwidth for node regeneration is equal to $\gamma = d\beta$. For the given distribution degree k, the MSR code has the smallest fragment size(α) and therefore minimizes the required storage bandwidth. The MSR fragment size and the MSR repair bandwidth parameter $\beta = \frac{\gamma}{d}$ are given by,

$$(\alpha_{MSR}, \gamma_{MSR}) = \left(\frac{\mathcal{M}}{k}, \frac{\mathcal{M}d}{k\,(d-k+1)} \right). \tag{3.1}$$

For MSR codes, $\alpha_{MSR} \leq \gamma_{MSR}$ 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

$$(\alpha_{MBR}, \gamma_{MBR}) = \left(\frac{2\mathcal{M}d}{2kd-k^2+k}, \frac{2\mathcal{M}d}{2kd-k^2+k} \right). \tag{3.2}$$

In our model, the system continuously monitors the redundancy level and initiates a repair when τ nodes are left within $\mathcal{A}$. The determination of τ, the type of repair (regeneration, reconstruction, or both) and the communication scheme for fragment retrieval (centralized or distributed) form a *threshold repair strategy*. We note that the practical implementation details of the redundancy monitoring and communication protocols for retrieving various fragments are beyond the scope of the present work. We focus on the theoretical aspects of the maintenance process.

3.3 Metrics

We evaluate the possible repair strategies using the following metrics.

DEFINITION 1 (Repair cost $c(\tau)$)**.** *The total cost $c(\tau)$ of fragment recovery when $\tau < n$ nodes remain in $\mathcal{A}$. It is defined as the number of bits that must be downloaded from the τ remaining nodes to restore the n encoded fragments in $\mathcal{A}$.*

DEFINITION 2 (Average repair cost per unit of time $r(\tau)$)**.** *The long-run average cost per unit of time ([42]) for maintaining fragments when repairing at τ (measured in bits per second).*

DEFINITION 3 (Mean time to data loss $MTTDL$)**.** *The mean time until $\mathcal{F}$ can no longer be recovered by nodes in $\mathcal{A}$ (i.e., more than k fragments are lost).*

DEFINITION 4 (Probability of data loss P_{DL})**.** *The probability of storing fewer than k fragments within $\mathcal{A}$.*

CHAPTER 4

DISTRIBUTED REPAIR

In this chapter, we analyze the distributed threshold repair strategy. Let τ denote the number of nodes remaining within $\mathcal{A}$ after the departure of $n-\tau$ nodes. We focus on determining the optimal repair threshold τ^*, which minimizes the average repair cost per unit of time. For our analysis, we assume that repairs occur instantaneously (we relax this assumption in Chapter 6).

4.1 Repair cost

In distributed repair, new nodes recover lost fragments by independently downloading existing fragments from other nodes. The repair process is initiated when a threshold of $k \leq \tau < n-1$ nodes remain within $\mathcal{A}$. During the repair, the $n-\tau$ lost fragments are restored at new nodes that exist or have entered $\mathcal{A}$. If $\tau \geq d$, this fragment recovery process can be performed through regeneration. Each of the $n-\tau$ replacement nodes downloads β symbols from d storage nodes and independently regenerates a lost fragment. If $\tau < d$, regeneration cannot be directly applied. To reduce the repair cost, we consider a dual scheme consisting of regeneration and reconstruction. First, $d-\tau$ nodes are repaired by downloading α symbols from k nodes and reconstructing $\mathcal{F}$. When d fragments become available, regeneration is applied to repair the remaining $n-d$ nodes. For each case, the repair cost is expressed as follows.

$$c(\tau) = \begin{cases} k\alpha(d-\tau) + \gamma(n-d), & \text{if } \tau < d \\ \gamma(n-\tau), & \text{if } \tau \geq d. \end{cases} \tag{4.1}$$

In (4.1), γ denotes the regeneration cost of a single fragment and depends on the regeneration code (see eqs. (3.1) and (**??**) for MSR and MBR). From (4.1), it is

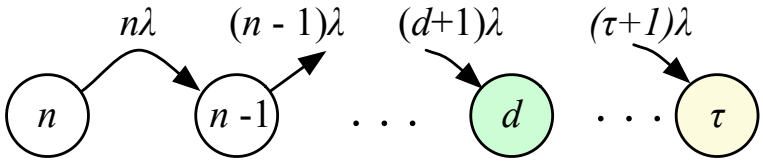

Figure 4.1: Markov chain model for a distributed threshold repair strategy.

evident that $c(\tau)$ monotonically decreases with τ. Moreover, the cost increase rate is higher when $\tau < d$. To determine the optimal threshold τ^*, we are interested in minimizing $r(\tau)$, which captures the long-term cost (for maintaining fragments) per unit of time. To calculate $r(\tau)$, we use the continuous-time Markov chain (CTMC) shown in Figure. 4.1, which models the node departure process (equivalently, the fragment loss process). The model consists of $n - \tau + 1$ states representing the number of fragments that remain within $\mathcal{A}$ after each node departure, until the fragments are repaired. The departure rate from a state i equals the node departure rate λ, times the number of nodes which store fragments at state i. Assume that whenever the CTMC is in state τ, it incurs cost at rate $c(\tau)$. Let $r(\tau, T)$ be the total expected cost in the interval $[0, T]$ [42]. The long-run cost rate starting from state i is then given by

$$r(\tau) = lim_{T\to\infty}\frac{r(\tau, T)}{T}. \tag{4.2}$$

For an irreducible CTMC with limiting distribution $\pi = [\pi_n, ..., \pi_\tau]$ the expected long-run cost rate at state τ becomes,

$$r(\tau) = r = \sum_{i=\tau}^{n} \pi_i c(i). \tag{4.3}$$

This is motivated by the idea that independent of the initial state, the CTMC spends a fraction of time p_i in state i where it incurs cost ar rate $c(i)$. Because the repair process is initiated only when state τ is reached, it follows that $c(i) = 0, \forall i \neq \tau$. Hence, eq. (4.3) degenerates to $r(\tau) = \pi_\tau c(\tau)$. To determine π_τ, we first derive the balance equations for the CTMC of Figure. 4.1 as follows (see Appendix

A for detailed derivation).

$$\pi_i = \begin{cases} \frac{n}{i}\pi_n, & \tau+1 \leq i \leq n-1 \\ n\lambda\pi_n, & i = \tau. \end{cases} \tag{4.4}$$

We use the normalization $\sum_i \pi_i = 1$ to compute π_n.

$$\pi_n = \frac{1}{1+n\lambda+nH_{n-1,\tau}}, \quad H_{n-1,\tau} = \sum_{i=1}^{n-1}\frac{1}{i} - \sum_{i=1}^{\tau}\frac{1}{i}. \tag{4.5}$$

In (4.5), $H_{n-1,\tau}$ denotes the difference between the harmonic numbers of $n-1$ and τ. Using (4.4), from which we obtain $\pi_\tau = n\lambda\pi_n$, and (4.5), we obtain the average cost $r(\tau) = \pi_\tau(\tau)$ as given by the following.

$$r(\tau) = \begin{cases} \frac{n\lambda(k\alpha(d-\tau)+\gamma(n-d))}{1+n\lambda+nH_{n-1,\tau}}, & \text{if } \tau < d \\ \frac{n\lambda(\gamma(n-\tau))}{1+n\lambda+nH_{n-1,\tau}}, & \text{if } \tau \geq d. \end{cases} \tag{4.6}$$

4.2 Optimal threshold

4.2.1 Regeneration

We use (4.6) to determine the optimal threshold τ^* which minimizes $r(\tau)$. This is given by the following propositions.

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

$$\tau^* = \begin{cases} d, & \lambda \leq \frac{H_{n-1,d}}{n-d-1} - \frac{1}{n}, \\ n-1, & \textit{otherwise.} \end{cases} \tag{4.7}$$

Proof. A straightforward minimization of $r(\tau)$ through differentiation involves harmonic sums. To determine τ^*, we compare $r(d)$ with the average repair cost at all other possible regeneration states $d+\delta$, where for δ satisfying $1 \leq \delta \leq n-d-1$, we check if $r(d) \leq r(d+\delta)$. Hence, consider:

$$r(d) \leq r(d+\delta). \tag{4.8}$$

Instead, we consider the above condition. On substituting $r(\tau)$ from (4.3),

$$\frac{n\lambda(n-d)\gamma}{1+n\lambda+nH_{n-1,d}} \leq \frac{n\lambda(n-d-\delta)\gamma}{1+n\lambda+nH_{n-1,d+\delta}} \Rightarrow$$

$$\lambda \leq \frac{(n-d)H_{d+\delta,d}}{\delta} - \frac{1}{n} - H_{n-1,d}. \tag{4.9}$$

Inequality (4.9) yields the maximum node departure rate λ for which it is more cost-efficient to repair at state $\tau = d$ than any other state $\tau = d + \delta$. We now examine the behavior of the right-hand side (RHS) of (4.9) as a function of δ for fixed n and d. The RHS of (4.9) has the same monotonicity as function

$$f(\delta) = \frac{H_{d+\delta,d}}{\delta} \approx \frac{\ln \frac{d+\delta}{d}}{\delta}, \tag{4.10}$$

because the rest of the terms do not depend on δ. In (4.10), we have approximated $H_n = \ln n + \epsilon$, where ϵ is the Euler-Mascheroni constant. This approximation holds for sufficiently large n. To find the monotonicity of $f(\delta)$ with respect to δ we compute the first derivative

$$f^{'}(\delta) = \frac{1}{\delta(d+\delta)} - \frac{ln(\frac{d+\delta}{d})}{\delta^2}. \tag{4.11}$$

Equating $f^{'}(\delta)$ with zero yields a log-linear function that cannot be solved analytically. We resort to the following bounds on the logarithmic function $ln(1+x)$ [43] to derive f's monotonicity.

$$\frac{2x}{2+x} \leq \ln(1+x), \quad \text{for } 0 \leq x < \infty. \tag{4.12}$$

By using the lower bound of $\ln(1+x)$, elementary calculations yield:

$$f'(\delta) \leq \frac{-1}{(\delta+d)(\delta+2d)} < 0, \quad \forall d, \delta > 0. \tag{4.13}$$

This proves that $f(\delta)$ is monotonically decreasing with δ. As a result, the departure rates λ for which (4.8) holds are also monotonically decreasing with δ. Substituting $\delta^* = n - d - 1$ to the RHS of (4.9) yields the maximum departure rate

$$\lambda \leq \frac{H_{n-1,d}}{n-d-1} - \frac{1}{n}, \tag{4.14}$$

for which $r(d) \leq r(d+\delta), \forall\delta$. In this case, minimization of $r(\tau)$ is achieved at $\tau^* = d$.

We now prove that for rates $\lambda > \frac{H_{n-1,d}}{n-d-1} - \frac{1}{n}$, the average cost $r(\tau)$ is minimized when $\tau = n-1$. Following a similar reasoning, we compare $r(\tau)$ at $\tau = n-1$ with $r(\tau)$ at any other possible regeneration threshold. We consider

$$r(n-1) \leq r(n-1-\delta), \tag{4.15}$$

where $1 \leq \delta \leq n-d-1$. By substituting $r(\tau)$ from (4.3), it follows that

$$\frac{1}{1+n\lambda} \leq \frac{\delta+1}{1+n\lambda+nH_{n-1,n-\delta-1}}.$$

$$\lambda \geq \frac{H_{n-1,n-\delta-1}}{\delta} - \frac{1}{n}. \tag{4.16}$$

Let

$$g(\delta) = \frac{H_{n-1,n-\delta-1}}{\delta} \approx \frac{\ln \frac{n-1}{n-\delta-1}}{\delta}. \tag{4.17}$$

Considering the derivative of (4.17) with respect to δ:

$$g'(\delta) = \frac{1}{\delta(n-\delta-1)} - \frac{\ln(n-1) - ln(n-\delta-1)}{\delta^2}. \tag{4.18}$$

We now utilize the upper bound $lnx < x-1,\ x > 1$ to derive the sign of $g'(\delta)$. From this bound, it follows that

$$\begin{aligned} \ln \frac{n-1}{n-\delta-1} &< \frac{n-1}{n-\delta-1} - 1 \Rightarrow \\ -\ln \frac{n-1}{n-\delta-1} &> \frac{-\delta}{n-\delta-1}. \end{aligned} \tag{4.19}$$

Substituting (4.19) to (4.18), we obtain that

$$\begin{aligned} g'(\delta) &> \frac{1}{\delta(n-\delta-1)} - \frac{\frac{\delta}{n-\delta-1}}{\delta^2} \Rightarrow \\ g'(\delta) &> 0. \end{aligned} \tag{4.20}$$

Hence, $g(\delta)$ is an increasing function of δ. The minimum λ for which $r(n-1) \leq r(n-1-\delta),\ \forall\delta$ is therefore obtained when $\delta^* = n-d-1$. Substituting $\delta = \delta^*$ to (4.16) completes the proof. □

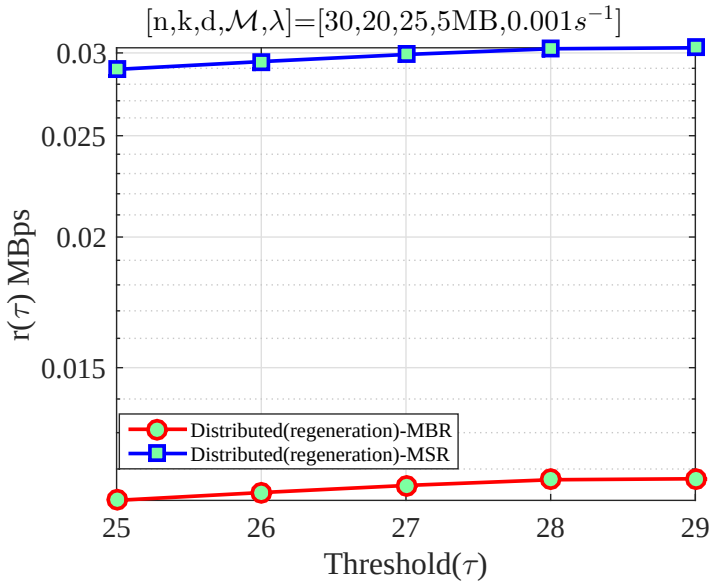

Figure 4.2: $r(\tau)$ vs. τ with $\tau^* = d$ for distributed repair(regeneration).

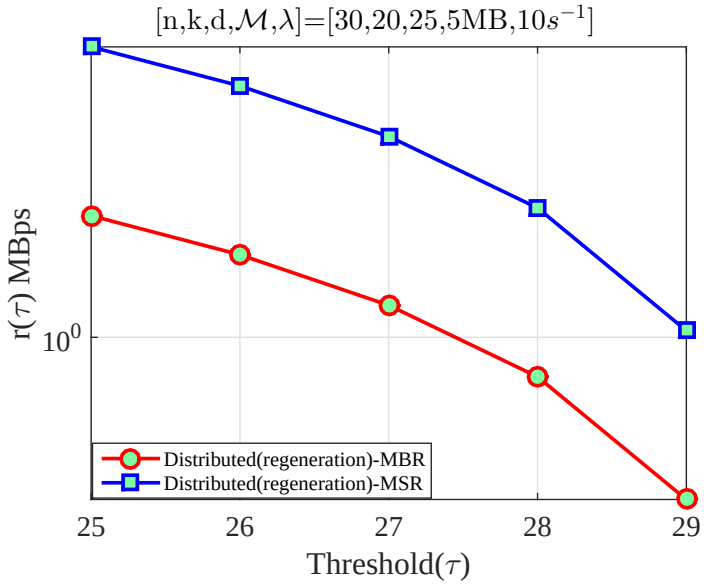

Figure 4.3: $r(\tau)$ vs. τ with $\tau^* = n - 1$ for distributed repair(regeneration).

Proposition 1 determines the departure rate regime for which repair at $\tau = d$, known as *lazy repair* [2], is more efficient than repairing at $\tau = n - 1$, known as

eager repair [2]. Figure 4.2 and 4.3 shows average repair cost per unit time as a function of threshold τ for different departure rate. Figure 4.2 verifies that for low λ, $r(\tau)$ is minimized at $\tau^* = d$. On the other hand, Figure 4.3 verifies that for high λ, the $r(\tau)$ is minimized at $\tau^* = n - 1$.

4.2.2 Regeneration plus reconstruction

We now examine if there is a departure rate regime for which reconstruction plus regeneration results in a lower average cost per unit of time compared to regeneration only. This rate regime is given by the following proposition.

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

$$\tau^* = \begin{cases} d, & \lambda \geq \frac{\gamma(n-d)H_{d,k}}{k\alpha(d-k)} - H_{n-1,d} - \frac{1}{n}, \\ k, & \textit{otherwise.} \end{cases} \tag{4.21}$$

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

$$r(d) \leq r(d - \delta). \tag{4.22}$$

On substituting for $r(\tau)$ using (4.3), we get:

$$\frac{n\lambda(n-d)\gamma}{1 + n\lambda + nH_{n-1,d}} \leq \frac{n\lambda(k\alpha\delta + (n-d)\gamma)}{1 + n\lambda + nH_{n-1,d-\delta}}. \tag{4.23}$$

$$\frac{n\lambda(n-d)\gamma}{1 + n\lambda + nH_{n-1,d}} \leq \frac{n\lambda(n-d)\gamma + k\alpha\delta}{1 + n\lambda + nH_{n-1,d} + nH_{d,d-\delta}}. \tag{4.24}$$

$$\text{Thus, } \lambda \geq \frac{(n-d)\gamma H_{d,d-\delta}}{k\alpha\delta} - \frac{1}{n} - H_{n-1,d}. \tag{4.25}$$

Expression (4.25) yields a bound on the minimum λ for which the optimal repair threshold is $\tau^* = d$. We now study the behavior of (4.25) as a function of δ for fixed n, k and d. Let,

$$h(\delta) = \frac{H_{d,d-\delta}}{\delta} \approx \frac{\ln \frac{d}{d-\delta}}{\delta}. \tag{4.26}$$

Considering the derivative of (4.26) with respect to δ:

$$h'(\delta) = \frac{1}{\delta(d-\delta)} - \frac{\ln(d) - ln(d-\delta)}{\delta^2}. \tag{4.27}$$

We now utilize the upper bound $lnx < x-1,\ x>1$, it follows that

$$\begin{aligned} \ln\frac{d}{d-\delta} &< \frac{n-1}{d-\delta} - 1 \Rightarrow \\ -\ln\frac{d}{d-\delta} &> \frac{-\delta}{d-\delta}. \end{aligned} \tag{4.28}$$

Substituting (4.28) to (4.27), we obtain that

$$\begin{aligned} h'(\delta) &> \frac{1}{\delta(d-\delta)} - \frac{\frac{\delta}{d-\delta}}{\delta^2} \Rightarrow \\ h'(\delta) &> 0. \end{aligned} \tag{4.29}$$

Therefore $h(\delta)$ is monotonically increasing with δ. Substituting the maximum $\delta^* = d-k$ yields the minimum departure rate,

$$\lambda \geq \frac{\gamma(n-d)H_{d,k}}{k\alpha(d-k)} - H_{n-1,d} - \frac{1}{n}. \tag{4.30}$$

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

We now prove that for rates $\lambda \leq \frac{\gamma(n-d)H_{d,k}}{k\alpha(d-k)} - H_{n-1,d} - \frac{1}{n}$, the average cost $r(\tau)$ per unit of time is minimized when $\tau = k$. We compare $r(\tau)$ at $\tau = k$ with $r(\tau)$ at any other possible repair threshold.

$$r(k) \leq r(k+\delta). \tag{4.31}$$

On substituting for $r(\tau)$ from (4.3), we get:

$$\begin{aligned} \frac{n\lambda(k\alpha(d-k)+\gamma(n-d))}{1+n\lambda+nH_{n-1,k}} &\leq \frac{n\lambda(k\alpha(d-k-\delta)+\gamma(n-d))}{1+n\lambda+nH_{n-1,k+\delta}} \\ \lambda &\leq \frac{(k\alpha(d-k)+\gamma(n-d))H_{k+\delta,k}}{k\alpha\delta} - \frac{1}{n} - H_{n-1,k} \end{aligned}$$

The above expression yields a bound on the maximum node departure rate λ for which the optimal repair threshold is $\tau^* = k$. We now study the behavior of (4.32) as a function of δ for fixed n, k and d. Let,

$$y(\delta) = \frac{H_{k+\delta,k}}{\delta} \approx \frac{\ln(k+\delta) - \ln k}{\delta}. \tag{4.32}$$

$$y^{'}(\delta) = \frac{1}{\delta(k+\delta)} - \frac{ln(\frac{k+\delta}{k})}{\delta^2}. \tag{4.33}$$

Equating $y^{'}(\delta)$ with zero yields a log-linear function and thus we consider the lower bound on logarithmic function in (4.33). By using the elementary calculations yield:

$$y'(\delta) \leq \frac{-1}{(\delta+k)(\delta+2k)} < 0, \quad \forall k, \delta > 0. \tag{4.34}$$

Since $k > 1$ it follows that $y(\delta)$ is also a monotonically decreasing function. Therefore, $\delta^* = d - k$ yields the maximum λ.

$$\lambda \leq \frac{(k\alpha(d-k) + \gamma(n-d))H_{d,k}}{k\alpha(d-k)} - H_{n-1,k} - \frac{1}{n} \tag{4.35}$$

$$\lambda \leq \frac{\gamma(n-d)H_{d,k}}{k\alpha(d-k)} - H_{n-1,d} - \frac{1}{n}, \tag{4.36}$$

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

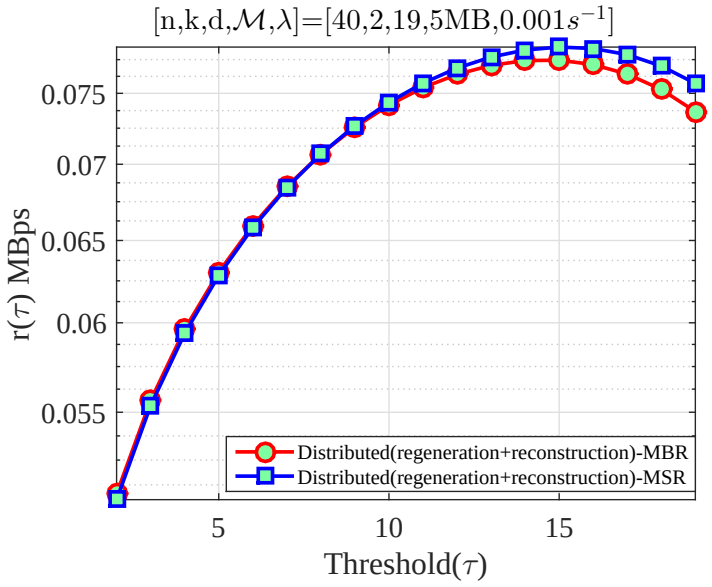

Figure 4.4: $r(\tau)$ vs. τ with $\tau^* = k$ for distributed repair(regeneration plus reconstruction).

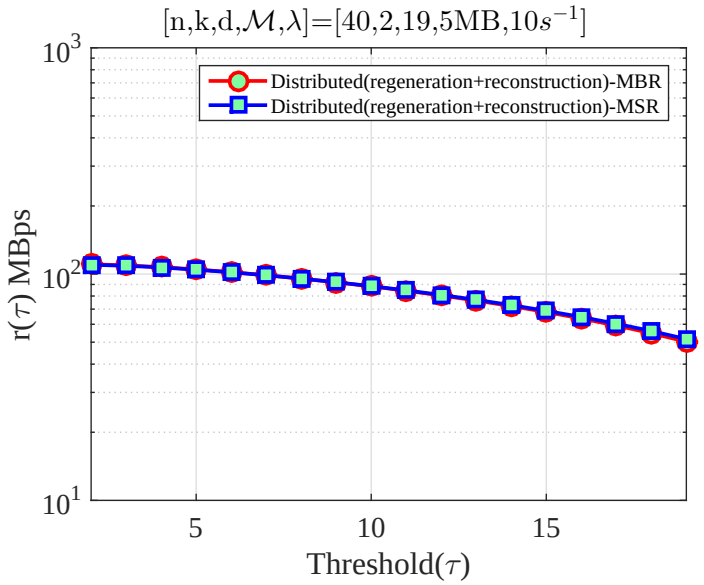

Figure 4.5: $r(\tau)$ vs. τ with $\tau^* = d$ for distributed repair(regeneration plus reconstruction).

Figures 4.4 and 4.5 show the average repair cost per unit time as a function of τ for $\tau \in [k, d]$ at different threshold departure rates. It can be seen that the analytical proof is in good terms with the plots. By combining Propositions 1 and 2, we can define the optimal repair strategy for any λ.

CHAPTER 5

CENTRALIZED REPAIR

In this chapter, centralized repair strategy is described. In the centralized repair strategy, repairs are performed by a *leader node* in two stages. In the first stage, the leader node downloads α symbols from k nodes and reconstructs $\mathcal{F}$. In the second stage, the leader node transmits α bits to each of the remaining $(n-\tau-1)$ nodes to restore the remaining $(n-\tau-1)$ fragments. The motivation behind this scheme is a possible reduction in the number of transmissions to carry out the repairs.

5.1 Repair cost

We investigate the optimal repair threshold τ^*, which minimizes the average repair cost per unit of time. The repair cost for a centralized repair strategy is then given by:

$$c(\tau) = \alpha\left(k+n-\tau-1\right). \tag{5.1}$$

The node departure process does not vary with the repair strategy. Therefore, the CTMC model shown in Fig. 5.1 applies for the centralized repair. According to (4.3), the long-run average repair cost per unit of time is given by:

$$r(\tau) = \pi_\tau c(\tau) = \frac{n\lambda\alpha\left(k+n-\tau-1\right)}{1+n\lambda+nH_{n-1,\tau}}. \tag{5.2}$$

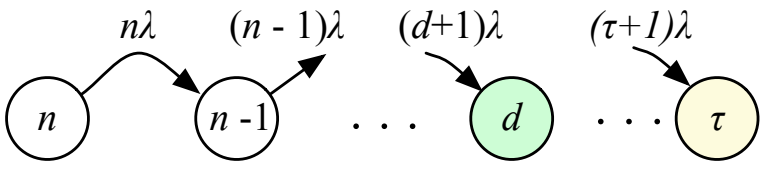

Figure 5.1: Markov chain model for a centralized threshold repair strategy.

5.2 Optimal threshold

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

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

$$\tau^* = \begin{cases} k, & \lambda \leq \frac{kH_{n-1,k}}{(n-k-1)} - \frac{1}{n}, \\ n-1, & \textit{otherwise.} \end{cases} \tag{5.3}$$

Proof. To determine τ^*, we compare $r(k)$ with other possible repair states $k+\delta$,

$$r(k) \leq r(k+\delta), \tag{5.4}$$

where $1 \leq \delta \leq n-k-1$. On substituting for $r(\tau)$ from (4.3), we obtain:

$$\begin{aligned} &\frac{n\lambda(k\alpha+\alpha(n-k-1))}{1+n\lambda+nH_{n-1,k}} \leq \frac{n\lambda(k\alpha+\alpha(n-k-\delta-1))}{1+n\lambda+nH_{n-1,k+\delta}}. \\ &\frac{k\alpha+\alpha(n-k-1)}{1+n\lambda+nH_{n-1,k}} \leq \frac{(k\alpha+\alpha(n-k-1))-\alpha\delta}{(1+n\lambda+nH_{n-1,k})-nH_{k+\delta,k}} \Rightarrow \\ &\lambda < \frac{(k\alpha+\alpha(n-k-1))H_{k+\delta,k}}{\alpha\delta} - \frac{1}{n} - H_{n-1,k}. \end{aligned} \tag{5.5}$$

Inequality (5.5) yields the maximum node departure rate λ 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 right-hand side (RHS) of (5.5) as a function of δ for fixed k and d. The RHS of (5.5) has the same monotonicity as function

$$f(\delta) = \frac{H_{k+\delta,k}}{\delta} \approx \frac{\ln\frac{k+\delta}{k}}{\delta}, \tag{5.6}$$

rate. On taking the derivative of $f(\delta)$ with respect to δ and approximating the logarithmic function using (4.12), we get:

$$f^{'}(\delta) = \frac{1}{\delta(\delta+k)} - \frac{ln\frac{\delta+k}{k}}{\delta^2} \Rightarrow \tag{5.7}$$

$$\tag{5.8}$$

Equating $f^{'}(\delta)$ with zero yields a log-linear function that cannot be solved analytically. We resort to the following bounds on the logarithmic function $ln(1+x)$ [43] to derive f's monotonicity.

$$\frac{2x}{2+x} \leq \ln(1+x), \quad \text{for } 0 \leq x < \infty. \tag{5.9}$$

By using the lower bound of $\ln(1+x)$, elementary calculations yield:

$$f'(\delta) \leq \frac{-1}{(\delta+d)(\delta+2d)} < 0, \quad \forall d, \delta > 0. \tag{5.10}$$

$$f^{'}(\delta) \quad \leq \quad \frac{-1}{(\delta+k)(\delta+2k)}. \tag{5.11}$$

As, the first derivative is negative, $f(\delta)$ is monotonically decreasing function for $\forall\delta$. Substituting $\delta^* = n-k-1$ to the RHS of (5.5) yields the maximum rate

$$\lambda \leq \frac{kH_{n-1,k}}{(n-k-1)} - \frac{1}{n}. \tag{5.12}$$

for which $r(k) \leq r(k+\delta), \forall\delta$. For this rate regime, 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 $\tau^* = n-1$.

$$r(n-1) \leq r(n-1-\delta). \tag{5.13}$$

where $1 \leq \delta \leq n-k-1$. On substituting for $r(\tau)$ from (4.3), we get:

$$\frac{n\lambda(k\alpha)}{1+n\lambda} \quad \leq \quad \frac{n\lambda(k\alpha+\alpha\delta)}{1+n\lambda+nH_{n-1,n-1+\delta}} \Rightarrow \tag{5.14}$$

$$\lambda \quad \geq \quad \frac{kH_{n-1,n-1-\delta}}{\delta} - \frac{1}{n}. \tag{5.15}$$

We determine the behavior Of the RHS of inequality (5.15) as a function of δ, we consider:

$$h(\delta) = \frac{ln(n-1) - ln(n-1-\delta)}{\delta} \tag{5.16}$$

Taking the first derivative of $h(\delta)$ with respect to δ.

$$h'(\delta) = \frac{-1}{\delta(\delta - n + 1)} - \frac{ln(-(n-1)/(\delta - n + 1))}{\delta^2}. \tag{5.17}$$

We utilize the upper bound $lnx < x - 1,\ x > 1$ to derive the sign of $h'(\delta)$. From this bound, it follows that

$$\begin{aligned} \ln \frac{n-1}{n-\delta-1} &< \frac{n-1}{n-\delta-1} - 1 \Rightarrow \\ -\ln \frac{n-1}{n-\delta-1} &> \frac{-\delta}{n-\delta-1}. \end{aligned} \tag{5.18}$$

Substituting (5.18) to (5.17), we obtain,

$$\begin{aligned} h'(\delta) &> \frac{1}{\delta(n-\delta-1)} - \frac{\frac{\delta}{n-\delta-1}}{\delta^2} \Rightarrow \\ h'(\delta) &> 0. \end{aligned} \tag{5.19}$$

Hence, $h(\delta)$ is an increasing function of δ. Substituting $\delta^* = n - k - 1$ yields the minimum rate for which $\tau^* = n - 1$.

$$\lambda \geq \frac{kH_{n-1,k}}{(n-k-1)} - \frac{1}{n}. \tag{5.20}$$

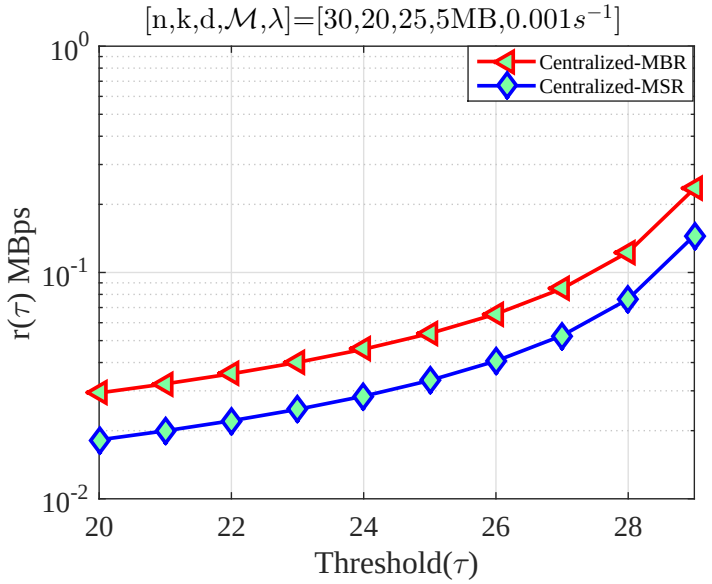

Figure 5.2: $r(\tau)$ vs. τ with $\tau^* = k$ for centralized repair.

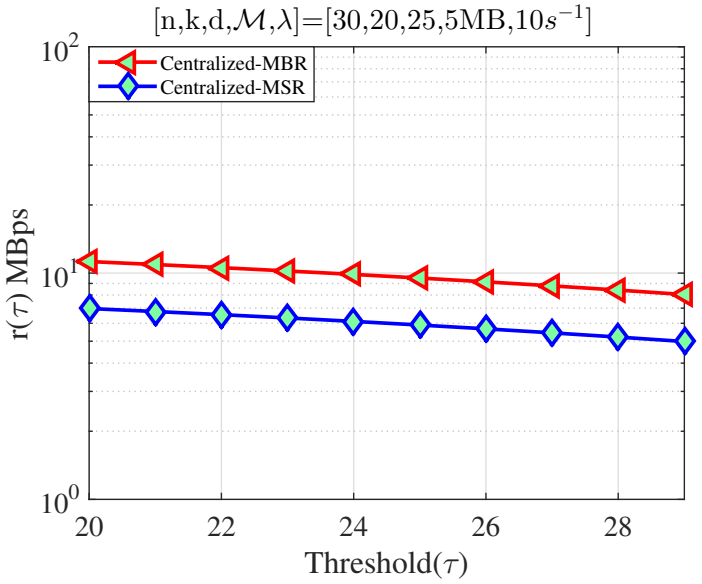

Figure 5.3: $r(\tau)$ vs. τ with $\tau^* = n - 1$ for centralized repair.

Figure 5.2 and 5.3 shows $r(\tau)$ as a function of the threshold τ for threshold departure rates as shown in proposition 3. Figure 5.2 Verifies that for low λ, $r(\tau)$ is minimized at $\tau^* = k$. On the other hand, Figure 5.3 verifies that for high λ, the $r(\tau)$ is minimized at $\tau^* = n - 1$. The graphical results comply with that obtained by analytical results.

CHAPTER 6

COMPARISON OF REPAIR STRATEGIES

In this chapter, we determine the rate 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 redundancy code parameters.

6.1 Eager Repair vs. Lazy Repair

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

$$\lambda_{low} \leq \min\left(\frac{H_{n-1,d}}{n-d-1} - \frac{1}{n}, \frac{\gamma(n-d)H_{d,k}}{k\alpha(d-k)} - H_{n-1,d} - \frac{1}{n}, \frac{kH_{n-1,k}}{(n-k-1)} - \frac{1}{n}\right)$$
$$\lambda_{high} > \max\left(\frac{H_{n-1,d}}{n-d-1} - \frac{1}{n}, \frac{\gamma(n-d)H_{d,k}}{k\alpha(d-k)} - H_{n-1,d} - \frac{1}{n}, \frac{kH_{n-1,k}}{(n-k-1)} - \frac{1}{n}\right).$$

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

$$\lambda_{low} \leq \min\left(\frac{H_{n-1,d}}{n-d-1} - \frac{1}{n}, \frac{\gamma(n-d)H_{d,k}}{k\alpha(d-k)} - H_{n-1,d} - \frac{1}{n}\right), \tag{6.1}$$
$$\lambda_{high} > \max\left(\frac{\gamma(n-d)H_{d,k}}{k\alpha(d-k)} - H_{n-1,d} - \frac{1}{n}, \frac{kH_{n-1,k}}{(n-k-1)} - \frac{1}{n}\right). \tag{6.2}$$

For any λ_{low}, the repair cost per unit of time is minimized when lazy repair is applied and the lowest possible repair threshold is selected. On the other hand,

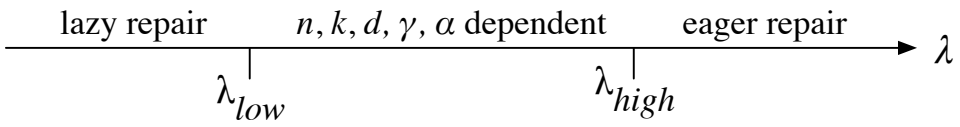

Figure 6.1: Optimal repair for different λ regimes.

for any λ_{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 rates do not lie in either of the λ regimes, then the optimal repair policy (eager vs. lazy) depends on the relationship of the code parameters and the repair strategy (centralized or distributed). The comparison of eager repair with lazy repair for different λ regimes is summarized in Fig. 6.1.

6.2 Centralized vs. Distributed repair

We now fix the departure rate λ and 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. Let $r(\tau)_d$ and $r(\tau)_c$ denote the average repair cost of distributed and centralized repair at τ, respectively. We consider the following inequality:

$$r(\tau)_d < r(\tau)_c. \tag{6.3}$$

In centralized repair, for fixed n, k, and d, $r(\tau)_c$ depends on α. As $\alpha_{MSR} \leq \alpha_{MBR}$, MSR codes minimize $r(\tau)_c$. Thus, we select MSR codes and centralized

repair for our comparison. Similarly, for given n, k, and d, the average repair cost $r(\tau)_d$ depends on the repair bandwidth γ. As $\gamma_{MBR} \leq \gamma_{MSR}$, MBR codes are selected to minimize $r(\tau)_d$. Substituting $r(\tau)_c$ and $r(\tau)_d$ from eqs. (1)-(4),

$$\begin{aligned} \frac{\mathcal{M}(2d)(n-\tau)\pi_\tau}{k(2d-k+1)} &< \frac{\mathcal{M}(n+k-\tau-1)\pi_\tau}{k} \Rightarrow \\ n+k-\tau-1 &< 2d. \end{aligned} \tag{6.4}$$

Inequality (6.4) determines the minimum number of surviving nodes for which MBR distributed repair emerges as the most cost-efficient strategy. The LHS of (6.4) is a decreasing function of τ. 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 (6.3), we obtain:

$$2d < n + k - \tau - 1. \tag{6.5}$$

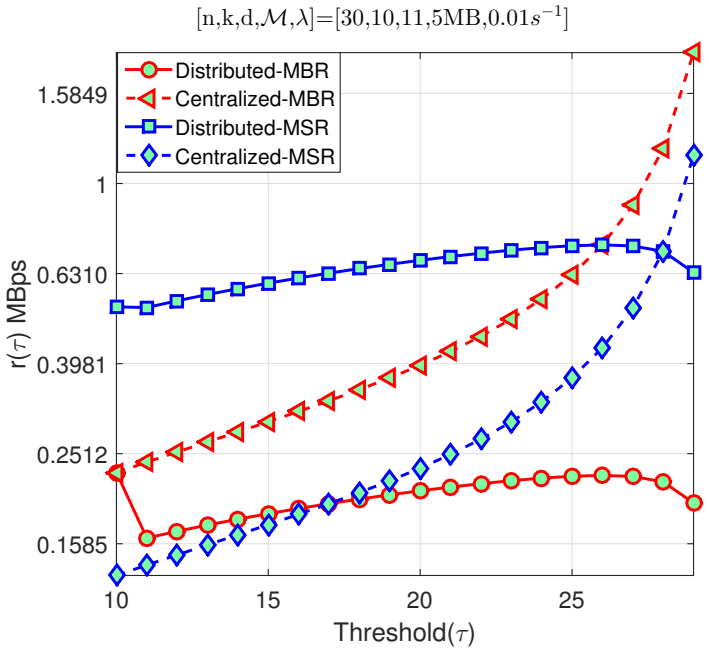

Figure 6.2: $r(\tau)$ vs. τ with $d < \frac{n+k-1}{3}$.

Minimizing the RHS of (6.5) 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 definition $d \geq k$. Therefore, there is no condition for which centralized MSR repair always outperforms distributed MBR repair. This can also be verified graphically and is as shown in Figure 6.2 and 6.3 . It can be seen that when $d < \frac{n+k-1}{3}$, no scheme is uniformly optimal for the given regime of τ. On the contrary, when $d > \frac{n+k-1}{3}$, distributed repair with MBR codes is uniformly optimal.

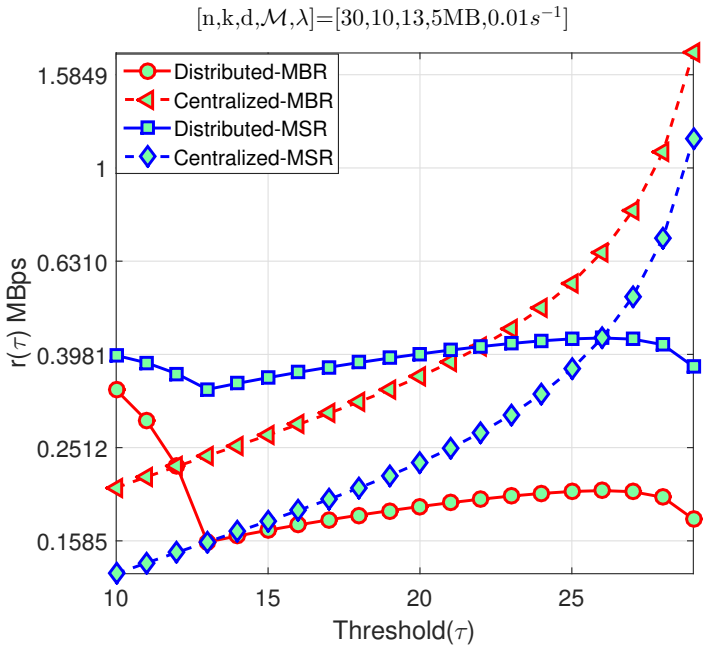

Figure 6.3: $r(\tau)$ vs. τ with $d > \frac{n+k-1}{3}$.

□

We now prove that if τ^* 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. To determine the optimal repair strategy, we compare $r(\tau)_c$ with $r(\tau)_d$ when $k \leq \tau^* < d$.

$$r(\tau)_c < r(\tau)_d.$$

Substituting $r(\tau)$ for distributed repair and centralized repair from (4.6) and (5.2), respectively, we obtain:

$$(\alpha(n+k-\tau-1))\pi_\tau < (\alpha(k(d-\tau)) + \gamma(n-d))\pi_\tau. \tag{6.6}$$

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 γ, we can write:

$$\begin{aligned} \alpha(n+k-\tau-1) &< \alpha(k(d-\tau)) + \alpha(n-d) \Rightarrow \\ k-1 &< k(d-\tau) - (d-\tau) \Rightarrow \\ k-1 &< (k-1)(d-\tau) \Rightarrow \\ \tau &< d. \end{aligned} \tag{6.7}$$

As $k \leq \tau^* < d$, inequality (6.7) 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 compared to MBR codes. Thus, centralized repair using MSR codes yields the optimal repair strategy. □

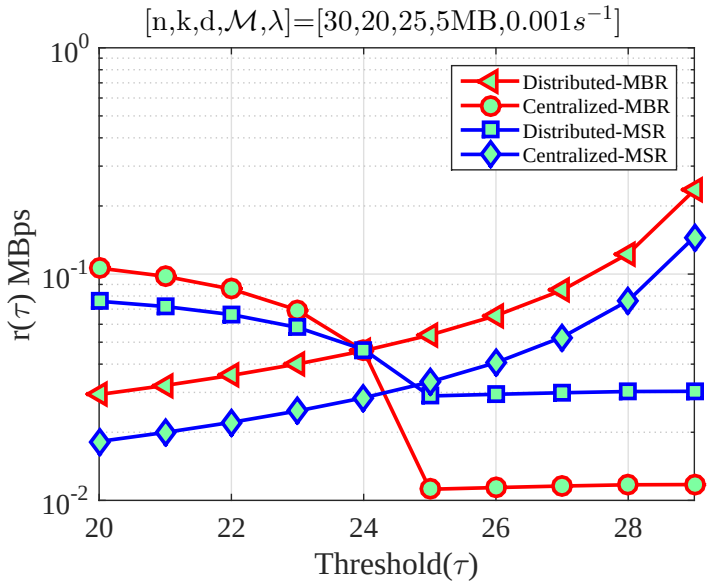

Figure 6.4: $r(\tau)$ vs. τ with $\tau^* = d$ for distributed repair.

The comparison of different repair strategies is shown in Figure. 6.4. Figure. 6.4 considers a low departure rate regime in which node departures are infrequent. For this regime, we observe that the cost of distributed repair is optimized by performing lazy repair ($\tau^* = d$).

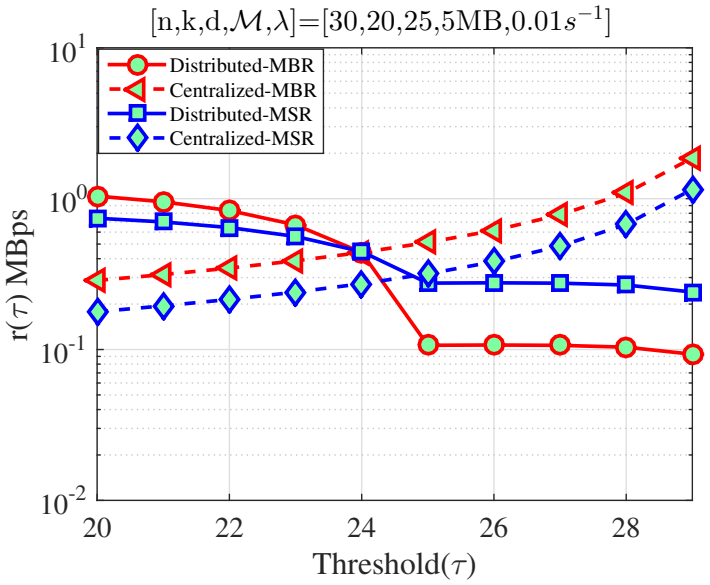

Figure 6.5: $r(\tau)$ vs. τ with $\tau^* = n - 1$ for distributed repair.

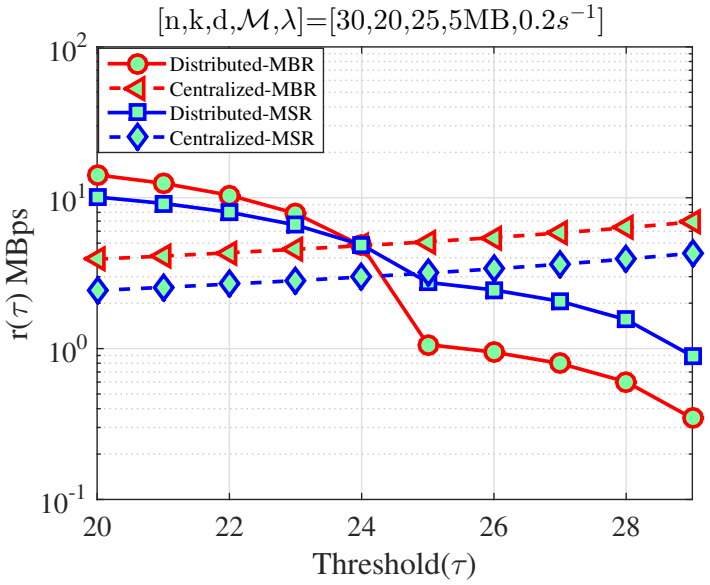

Figure 6.6: $r(\tau)$ vs. τ for $\tau^* = k$ for centralized repair.

When we move to the λ_{high} regime (Figures 6.5,6.7), eager repair ($\tau^* = n - 1$) becomes optimal. Moreover, as $d > \frac{n+k-1}{3}$, distributed repair using MBR codes

results in the best performance when $d \leq \tau \leq n-1$. On the other hand, for $k \leq \tau < d$, centralized repair using MSR codes outperforms all other strategies. Overall, *distributed repair using MBR codes results in the lowest average repair cost.*

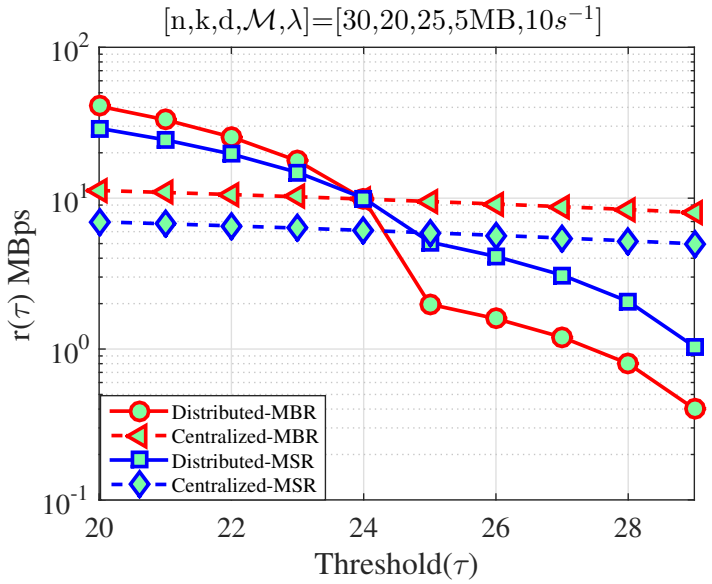

Figure 6.7: $r(\tau)$ vs. τ for $\tau^* = n-1$ for centralized repair.

CHAPTER 7

EXTENDED MODEL WITH INCOMPLETE REPAIRS

In the previous chapters, we have assumed that repairs occur with a very high repair rate (almost instantaneously) relative to the node to node departure process. In a realistic scenario, the rate of repair is limited by the available communication bandwidth B. The delay in repairing the lost fragments can lead to permanent loss of $\mathcal{F}$, if fewer than k nodes remain within $\mathcal{A}$ before the file repair is completed.

For this realistic scenario, we calculate P_{DL} and $MTTDL$ using the CTMC model shown in Figure. 7.1. The model consists of $n-k+1$ possible states with each state representing the number of fragments that remain within $\mathcal{A}$. In this model, repair is performed only at state τ, but with rate μ. The model in Figure. 7.1 assumes that all repairs proceed in parallel and that no repaired fragments are available until all repairs are completed. This model faithfully reflects the distributed repair process in which repairs at each node proceed independently using shared resources. The treatment of other repair policies such as sequential repair, follows a similar analysis.

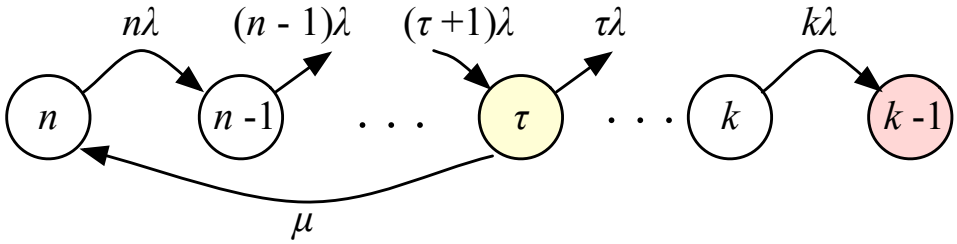

Figure 7.1: Markov chain model under incomplete repairs.

7.1 Probability of Data Loss

For the CTMC model in Figure. 7.1, state $k-1$ is an absorbing state, as $\mathcal{F}$ can no longer be reconstructed if fewer than k fragments remain in $\mathcal{A}$. Thus, the

probability of permanent loss of $\mathcal{F}$ equals the steady-state probability of reaching state $k-1$. That is, $P_{DL} = \pi_{k-1}$. To determine π_{k-1}, we write the balance equations as follows (see Appendix A for details).

$$\pi_i = \begin{cases} \frac{n}{i}\pi_n, & \tau+1 \leq i \leq n-1 \\ \frac{n\lambda}{\tau\lambda+\mu}\pi_n, & i = \tau \\ \frac{\tau}{i}\pi_\tau\pi_n, & k \leq i \leq \tau-1 \\ \tau\lambda\pi_\tau\pi_n, & i = k-1 \end{cases} \tag{7.1}$$

We use the normalization $\sum_i \pi_i = 1$ to compute π_n and express the probability of data loss as:

$$P_{DL} = \frac{\frac{\tau\lambda(n\lambda)}{\tau\lambda+\mu}}{1 + nH_{n-1,\tau} + \frac{n\lambda}{\tau\lambda+\mu}(1 + \tau H_{\tau-1,k-1} + \tau\lambda)}.$$

7.2 Mean Time to Data Loss

Another typical reliability metric for storage systems is the $MTTDL$. Based on the CTMC of Figure. 7.1, we determine the expected hitting time of state $k-1$ using the theorem presented in [28]. The theorem states that for a CTMC $X(t)$ with rate matrix Q and J being a subset of the state space, the vector of expected hitting times $k^J = (k_i^J : i \in S)$ is the minimal non-negative solution to the system of linear equations.

$$\begin{cases} k_i^J = 0, & \text{for } i \in J \\ -\sum_{l \in S} q_{il} k_l^J = 1, & \text{for } i \notin J. \end{cases} \tag{7.2}$$

In our context, $J = \{k-1\}$ and the rate matrix Q is a $n-k+1 \times n-k+1$ square matrix and is given by:

$$Q = \begin{pmatrix} -n\lambda & n\lambda & \dots & 0 \\ 0 & -(n-1)\lambda & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ \frac{B}{c(\tau)} & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & \dots & -k\lambda & k\lambda \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

Thus, the $MTTDL$ is given by the last term of the vector k_J. A closed-form solution cannot be analytically obtained for k_J. However, given the system parameters, the $MTTDL$ can be computed using numerical methods.

7.3 Tradeoff Between System Reliability and Repair Cost

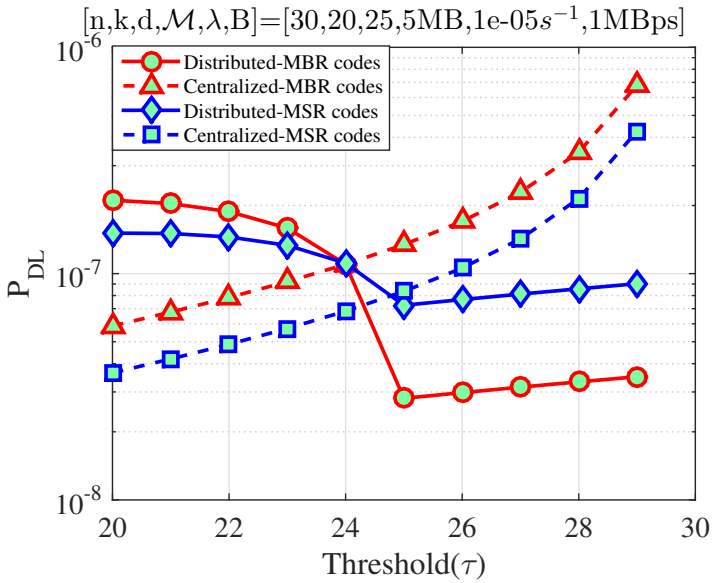

Figure 7.2: Probability of data loss as a function of τ at λ_{low}.

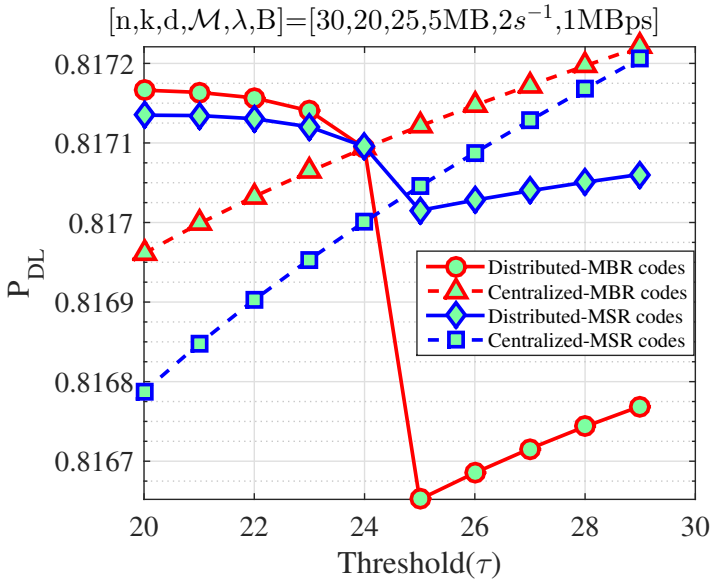

Figure 7.3: Probability of data loss as a function of τ at λ_{high}.

Figures 7.2, 7.3 and 7.4, 7.5 show P_{DL} and $MTTDL$ as a function of the repair threshold τ, respectively. It can be observed that system reliability is the least when repair threshold is set to $\tau = n - 1$. This can be attributed to the fact that if repair threshold is set to a high value, then on moving to state $\tau - 1$, the system has no chance being able to repair, thus increasing chances of data loss. But on the contrary, if the threshold τ is set to a low value, then assuming that repair is faster than departure rate, the time it takes to repair threshold state τ is high. Hence, $MTTDL$ is increases as τ decreases. We observe that both P_{DL} and $MTTDL$ are optimized when $\tau = d$. At a low rate regime λ_{low}, setting $\tau = d$ also optimizes the average repair cost per unit of time. However, at a high rate regime λ_{high} a tradeoff is established between file maintenance and file reliability. The eager repair strategy, which is shown to minimize the average repair cost per unit of time under this scenario, has an increased P_{DL} and reduced $MTTDL$ as compared to reliability-optimal values for these parameters. Thus, depending on the parameter of interest, the threshold for repair should be chosen accordingly.

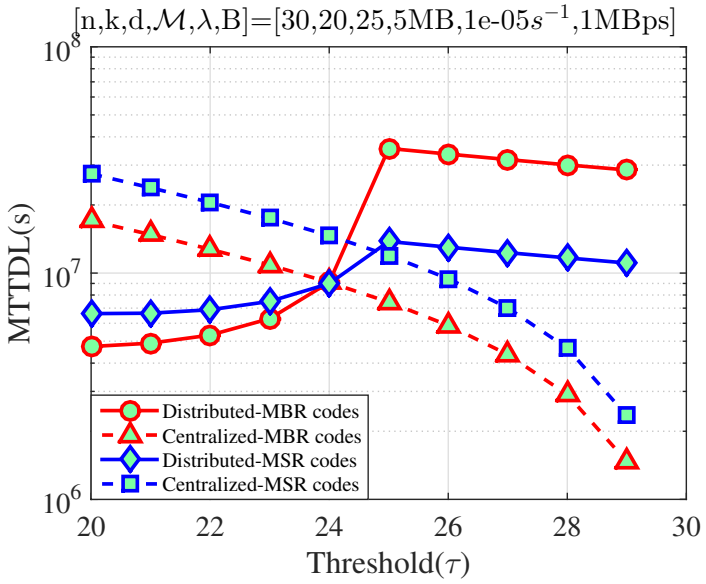

Figure 7.4: MTTDL as a function of τ at λ_{low}.

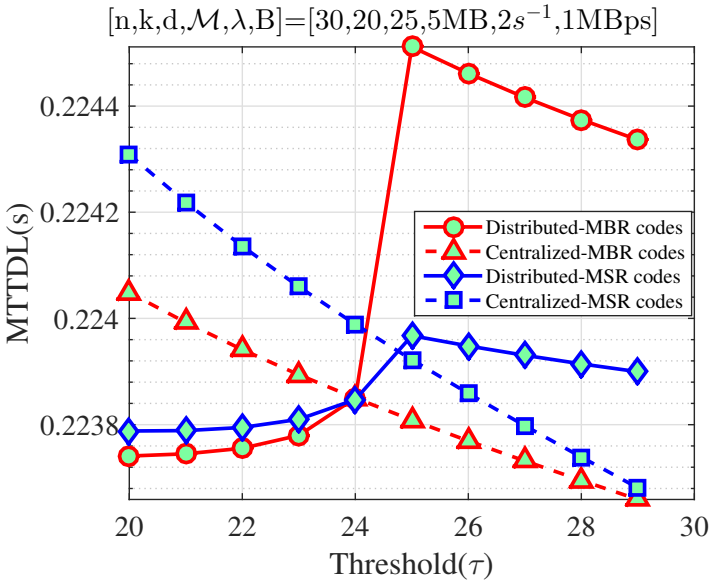

Figure 7.5: MTTDL as a function of τ at λ_{high}.

CHAPTER 8

CONCLUSION

In this thesis, we studied the data reliability problem for a community of devices forming a mobile cloud storage system. We focused our analysis on threshold-based maintenance using regeneration. We analyzed two repair strategies, namely distributed and centralized repair. For each strategy, we derived the optimal repair threshold that minimizes the repair bandwidth as a function of the node departure rate. The latter also signifies the fragment loss rate. We showed that the optimal repair threshold and strategy depends on many system parameters. Our results showed that no one strategy is optimal for all possible system configurations. For high-mobility scenarios, eager repair is the optimal repair policy. For low-mobility scenarios, lazy repair yields a lower repair cost. For relatively static networks, applying reconstruction first and then resorting to regeneration becomes the optimal . We also determine the optimal repair strategy for a given departure rate.

Finally, we analyzed the storage system reliability when repairs can be incomplete due to communication bandwidth constraints. Specifically, we determined the probability of data loss and the mean time to data loss as a function of the repair threshold and other system parameters. We also discussed the tradeoff associated with system reliability metrics and average cost per time. Our findings showed that in high-mobility scenarios, a tradeoff is established between file reliability and the average cost for maintaining a file.

APPENDIX A

APPENDIX

Derivation of Steady State Probabilities

Let π_i denote the steady-state probability that the chain is in state i. Under steady-state condition, the probability at each node can be written as:

$$\begin{aligned}
\text{At node (n-1), } n\lambda\pi_n &= (n-1)\lambda\pi_{n-1} \\
\Rightarrow \pi_{n-1} &= \frac{n}{n-1}\pi_n \\
\text{At node (n-2), } (n-1)\,\lambda\pi_{n-1} &= (n-2)\lambda\pi_{n-2} \\
\Rightarrow \pi_{n-2} &= \frac{n}{n-2}\pi_n \\
&\vdots \\
\text{At node } \tau+1,\ \pi_{\tau+1} &= \frac{n}{n-(n-\tau-1)}\pi_n \\
\text{At node } \tau,\ \pi_\tau &= (\tau+1)\lambda\pi_{\tau+1} \\
\Rightarrow \pi_\tau &= n\lambda\pi_n
\end{aligned}$$

$$\text{Since, } \sum_{i=1}^{n-\tau+1}\pi_i = 1$$

$$\sum_{i=1}^{n-\tau-1}\tfrac{n}{n-i}\pi_n + \underbrace{n\lambda\pi_n}_{\pi_\tau} + \pi_n = 1$$

$$\begin{aligned}
\Rightarrow \pi_n &= \frac{1}{1+n\lambda+\sum_{i=1}^{n-\tau-1}\frac{n}{n-i}} \\
\text{Also, } \pi_\tau &= n\lambda\pi_n \\
&= \frac{n\lambda}{1+n\lambda+\sum_{i=1}^{n-\tau-1}\frac{n}{n-i}} \\
&= \frac{n\lambda}{1+n\lambda+n\sum_{k=\tau+1}^{n-1}\frac{1}{k}} \\
&= \frac{n\lambda}{1+n\lambda+n\,(H_{n-1}-H_\tau)}
\end{aligned}$$

Derivation of Probability of data loss

Let π_i denote the steady-state probability that the chain is in state i. Under steady-state condition, the probability at each node can be written as:

$$\text{At node (n-1), } n\lambda\pi_n = (n-1)\lambda\pi_{n-1}$$
$$\Rightarrow \pi_{n-1} = \frac{n}{n-1}\pi_n$$
$$\text{At node (n-2), } (n-1)\,\lambda\pi_{n-1} = (n-2)\lambda\pi_{n-2}$$
$$\Rightarrow \pi_{n-2} = \frac{n}{n-2}\pi_n$$
$$\vdots$$
$$\text{At node } \tau+1, \pi_{\tau+1} = \frac{n}{n-(n-\tau-1)}\pi_n$$
$$\text{At node } \tau, (\tau\lambda + n)\pi_\tau = (\tau+1)\lambda\pi_{\tau+1}$$
$$\Rightarrow \pi_\tau = \frac{n\lambda}{\tau\lambda+\mu}\pi_n$$
$$\text{At node } \tau-1, ((\tau-1)\lambda + n)\pi_{\tau-1} = (\tau)\lambda\pi_\tau$$
$$\Rightarrow \pi_{\tau-1} = \frac{\tau}{\tau-1}\frac{n\lambda}{\tau\lambda+\mu}\pi_n$$
$$\vdots$$
$$\text{At node } k, \pi_k = \frac{\tau}{.}k\frac{n\lambda}{\tau\lambda+\mu}\pi_n$$
$$\text{At node } k-1, \pi_{k-1} = k\lambda\pi_k$$
$$\Rightarrow \pi_{k-1} = \tau\lambda\frac{n\lambda}{\tau\lambda+\mu}\pi_n$$

$$\text{Since, } \sum_{i=1}^{n-\tau+1}\pi_i = 1$$

$$\pi_n = \frac{1}{1 + n\,(H_{n-1} - H_\tau) + \frac{n\lambda}{\tau\lambda+\mu}(1 + \tau H_{\tau-1,k-1} + \tau\lambda)}$$

$$\text{Thus, } P_{DL} = \pi_{k-1} = \frac{\tau\lambda\frac{n\lambda}{\tau\lambda+\mu}\pi_n}{1+n(H_{n-1}-H_\tau)+\frac{n\lambda}{\tau\lambda+\mu}(1+\tau H_{\tau-1,k-1}+\tau\lambda)}$$

REFERENCES

[1] M. Belleschi, G. Fodor, and A. Abrardo. Performance analysis of a distributed resource allocation scheme for d2d communications. In *GLOBECOM Workshops (GC Wkshps), 2011 IEEE*, pages 358–362. IEEE, 2011.

[2] R. Bhagwan, K. Tati, Y.-C. Cheng, S. Savage, and G. M. Voelker. Total recall: System support for automated availability management. In *NSDI*, volume 4, pages 25–25, 2004.

[3] B. Calder, J. Wang, A. Ogus, N. Nilakantan, A. Skjolsvold, S. McKelvie, Y. Xu, S. Srivastav, J. Wu, H. Simitci, et al. Windows azure storage: a highly available cloud storage service with strong consistency. In *Proceedings of the Twenty-Third ACM Symposium on Operating Systems Principles*, pages 143–157. ACM, 2011.

[4] Y. Chen, S. Jain, V. K. Adhikari, Z.-L. Zhang, and K. Xu. A first look at inter-data center traffic characteristics via yahoo! datasets. In *INFOCOM, 2011 Proceedings IEEE*, pages 1620–1628. IEEE, 2011.

[5] B.-G. Chun, F. Dabek, A. Haeberlen, E. Sit, H. Weatherspoon, M. F. Kaashoek, J. Kubiatowicz, and R. Morris. Efficient replica maintenance for distributed storage systems. In *NSDI*, volume 6, pages 4–4, 2006.

[6] Cisco. The zettabyte eratrends and analysis, 2014.

[7] L. P. Cox, C. D. Murray, and B. D. Noble. Pastiche: Making backup cheap and easy. *ACM SIGOPS Operating Systems Review*, 36(SI):285–298, 2002.

[8] F. Dabek, M. F. Kaashoek, D. Karger, R. Morris, and I. Stoica. Wide-area cooperative storage with cfs. *ACM SIGOPS Operating Systems Review*, 35(5):202–215, 2001.

[9] A. G. Dimakis, P. B. Godfrey, Y. Wu, M. J. Wainwright, and K. Ramchandran. Network coding for distributed storage systems. *IEEE Transactions on Information Theory*, 56(9):4539–4551, 2010.

[10] A. Duminuco and E. Biersack. A practical study of regenerating codes for peer-to-peer backup systems. In *Distributed Computing Systems, 2009. ICDCS'09. 29th IEEE International Conference on*, pages 376–384. IEEE, 2009.

[11] 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 *OSDI*, pages 61–74, 2010.

[12] J. Gantz and D. Reinsel. The digital universe in 2020: Big data, bigger digital shadows, and biggest growth in the far east-united states, Feb. 2013.

[13] B. Gastón, J. Pujol, and M. Villanueva. Quasi-cyclic minimum storage regenerating codes for distributed data compression. In *Data Compression Conference (DCC), 2011*, pages 33–42. IEEE, 2011.

[14] S. Genelius. The data explosion in 2014 minute by minute infographic, 2014.

[15] S. Ghemawat, H. Gobioff, and S.-T. Leung. The google file system. In *ACM SIGOPS operating systems review*, volume 37, pages 29–43. ACM, 2003.

[16] F. Giroire, J. Monteiro, and S. Pérennes. Peer-to-peer storage systems: a practical guideline to be lazy. In *Global Telecommunications Conference (GLOBECOM 2010), 2010 IEEE*, pages 1–6. IEEE, 2010.

[17] N. Golrezaei, A. G. Dimakis, and A. F. Molisch. Device-to-device collaboration through distributed storage. In *Global Communications Conference (GLOBECOM), 2012 IEEE*, pages 2397–2402. IEEE, 2012.

[18] Y. S. Han, R. Zheng, and W. H. Mow. Exact regenerating codes for byzantine fault tolerance in distributed storage. In *INFOCOM, 2012 Proceedings IEEE*, pages 2498–2506. IEEE, 2012.

[19] Y. Hu, P. P. Lee, and K. W. Shum. Analysis and construction of functional regenerating codes with uncoded repair for distributed storage systems. In *INFOCOM, 2013 Proceedings IEEE*, pages 2355–2363. IEEE, 2013.

[20] Y. Hu, Y. Xu, X. Wang, C. Zhan, and P. Li. Cooperative recovery of distributed storage systems from multiple losses with network coding. *Selected Areas in Communications, IEEE Journal on*, 28(2):268–276, 2010.

[21] S. Jiekak, A.-M. Kermarrec, N. Le Scouarnec, G. Straub, and A. Van Kempen. Regenerating codes: A system perspective. *ACM SIGOPS Operating Systems Review*, 47(2):23–32, 2013.

[22] B. Kaufman and B. Aazhang. Cellular networks with an overlaid device to device network. In *Signals, Systems and Computers, 2008 42nd Asilomar Conference on*, pages 1537–1541. IEEE, 2008.

[23] A. Kiani and S. Akhlaghi. Selective regenerating codes. *Communications Letters, IEEE*, 15(8):854–856, 2011.

[24] J. Kubiatowicz, D. Bindel, Y. Chen, S. Czerwinski, P. Eaton, D. Geels, R. Gummadi, S. Rhea, H. Weatherspoon, W. Weimer, et al. Oceanstore: An architecture for global-scale persistent storage. *ACM Sigplan Notices*, 35(11):190–201, 2000.

[25] M. Landers, H. Zhang, and K.-L. Tan. Peerstore: Better performance by relaxing in peer-to-peer backup. In *Peer-to-Peer Computing, 2004. Proceedings. Proceedings. Fourth International Conference on*, pages 72–79. IEEE, 2004.

[26] V. Lenders, G. Karlsson, and M. May. Wireless ad hoc podcasting. In *Sensor, Mesh and Ad Hoc Communications and Networks, 2007. SECON'07. 4th Annual IEEE Communications Society Conference on*, pages 273–283. IEEE, 2007.

[27] M. Lillibridge, S. Elnikety, A. Birrell, M. Burrows, and M. Isard. A cooperative internet backup scheme. In *Proceedings of the annual conference on USENIX Annual Technical Conference*, pages 3–3. USENIX Association, 2003.

[28] J. R. Norris. Markov chains, cambridge series in statistical and probabilistic mathematics, vol. 2, 1998.

[29] J. Ott and M. J. Pitkanen. Dtn-based content storage and retrieval. In *World of Wireless, Mobile and Multimedia Networks, 2007. WoWMoM 2007. IEEE International Symposium on a*, pages 1–7. IEEE, 2007.

[30] J. Paakkonen, P. Dharmawansa, C. Hollanti, and O. Tirkkonen. Distributed storage for proximity based services. In *Communication Technologies Workshop (Swe-CTW), 2012 Swedish*, pages 30–35. IEEE, 2012.

[31] J. Paakkonen, C. Hollanti, and O. Tirkkonen. Device-to-device data storage for mobile cellular systems. In *Globecom Workshops (GC Wkshps), 2013 IEEE*, pages 671–676. IEEE, 2013.

[32] J. Pääkkönen, C. Hollanti, and O. Tirkkonen. Device-to-device data storage for mobile cellular systems. *arXiv preprint arXiv:1309.6123*, 2013.

[33] J. Pääkkönen, C. Hollanti, and O. Tirkkonen. Device-to-device data storage with regenerating codes. *arXiv preprint arXiv:1411.1608*, 2014.

[34] M. J. Pitkänen and J. Ott. Redundancy and distributed caching in mobile dtns. In *Proceedings of 2nd ACM/IEEE international workshop on Mobility in the evolving internet architecture*, page 8. ACM, 2007.

[35] J. S. Plank. T1: erasure codes for storage applications. In *Proc. of the 4th USENIX Conference on File and Storage Technologies*, pages 1–74, 2005.

[36] K. Rashmi, N. B. Shah, P. V. Kumar, and K. Ramchandran. Explicit construction of optimal exact regenerating codes for distributed storage. In *Communication, Control, and Computing, 2009. Allerton 2009. 47th Annual Allerton Conference on*, pages 1243–1249. IEEE, 2009.

[37] K. V. Rashmi, N. B. Shah, and P. V. Kumar. Optimal exact-regenerating codes for distributed storage at the msr and mbr points via a product-matrix construction. *Information Theory, IEEE Transactions on*, 57(8):5227–5239, 2011.

[38] S. C. Rhea, P. R. Eaton, D. Geels, H. Weatherspoon, B. Y. Zhao, and J. Kubiatowicz. Pond: The oceanstore prototype. In *FAST*, volume 3, pages 1–14, 2003.

[39] R. Rodrigues and B. Liskov. High availability in dhts: Erasure coding vs. replication. In *Peer-to-Peer Systems IV*, pages 226–239. 2005.

[40] A. Rowstron and P. Druschel. Storage management and caching in past, a large-scale, persistent peer-to-peer storage utility. In *ACM SIGOPS Operating Systems Review*, volume 35, pages 188–201. ACM, 2001.

[41] N. B. Shah, K. Rashmi, P. V. Kumar, and K. Ramchandran. Distributed storage codes with repair-by-transfer and nonachievability of interior points on the storage-bandwidth tradeoff. *Information Theory, IEEE Transactions on*, 58(3):1837–1852, 2012.

[42] H. C. Tijms. *A first course in stochastic models*. John Wiley and Sons, 2003.

[43] F. Topsok. Some bounds for the logarithmic function. *Inequality theory and applications*, 4:137, 2006.

[44] G. Utard and A. Vernois. Data durability in peer to peer storage systems. In *Cluster Computing and the Grid, 2004. CCGrid 2004. IEEE International Symposium on*, pages 90–97. IEEE, 2004.

[45] A. Wang and Z. Zhang. Exact cooperative regenerating codes with minimum-repair-bandwidth for distributed storage. In *INFOCOM, 2013 Proceedings IEEE*, pages 400–404. IEEE, 2013.

[46] H. Weatherspoon and J. D. Kubiatowicz. Erasure coding vs. replication: A quantitative comparison. In *Peer-to-Peer Systems*, pages 328–337. 2002.

[47] Y. Wu, A. G. Dimakis, and K. Ramchandran. Deterministic regenerating codes for distributed storage. In *Allerton Conference on Control, Computing, and Communication*. Citeseer, 2007.

[48] C.-H. Yu, O. Tirkkonen, K. Doppler, and C. Ribeiro. On the performance of device-to-device underlay communication with simple power control. In *Vehicular Technology Conference, 2009. VTC Spring 2009. IEEE 69th*, pages 1–5. IEEE, 2009.