Relations Between Entity Sizes and Error-Correction Coding Codewords and Effective Data Loss

Ilias Iliadis
IBM Research Europe – Zurich
8803 Rüschlikon, Switzerland
email: ili@zurich.ibm.com

***Abstract*—Erasure-coding redundancy schemes are employed in storage systems to cope with device and component failures. Data durability is assessed by the Mean Time to Data Loss** (MTTDL) **and the Expected Annual Fraction of Entity Loss** (EAFEL) **reliability metrics. In particular, the** EAFEL **metric assesses losses at an entity, say file, object, or block level. This metric is affected by the number of codewords that entities span. The distribution of this number is obtained analytically as a function of the size of the entities and the frequency of their occurrence. The deterministic and the random entity placement cases are investigated. It is established that for certain deterministic placements of variable-size entities, the distribution of the number of codewords that entities span also depends on the actual entity placement. To evaluate the durability of storage systems in the case of variable-size entities, we introduce the Expected Annual Fraction of Effective Data Loss** (EAFEDL) **reliability metric, which assesses the fraction of stored user data that is lost by the system annually at the entity level. The** MTTDL**,** EAFEL**, and** EAFEDL **metrics are assessed analytically for erasure-coding redundancy schemes and for the clustered, declustered, and symmetric data placement schemes. These metrics are derived in closed-form for the case of lazy rebuilds and in the presence of correlated latent symbol errors. It is demonstrated that an increased variability of entity sizes results in improved** EAFEL**, but degraded** EAFEDL**. It is established that both reliability metrics are adversely affected by the size of the erasure-coding symbols. The** EAFEL **and** EAFEDL **reliability metrics are evaluated for some real-world erasure coding schemes employed by enterprises. The analytical reliability expressions derived can identify efficient erasure coding schemes and can be used to dimension and provision storage systems to provide desired levels of durability.**

Keywords–Storage; Reliability analysis; MTTDL; EAFDL; EAFEL; EAFEDL; MDS codes; Unrecoverable or latent symbol errors; Deferred recovery or repair; stochastic modeling.

I. INTRODUCTION

The durability of data storage systems and cloud offerings is affected by device and component failures [1]. Desired reliability levels are ensured by employing erasure-coding redundancy schemes for recovering lost data [2-5].

The frequency of data loss events is assessed by the Mean Time to Data Loss (MTTDL) metric that has been widely used to assess the reliability of storage systems [4][5]. Also, the amount of data loss is obtained by the Expected Annual Fraction of Data Loss (EAFDL) metric that was introduced in [6]. This metric was recently complemented by the Expected Annual Fraction of Entity Loss (EAFEL) metric [7]. The EAFEL metric assesses data losses at an *entity*, say file, object, or block level, whereas the EAFDL metric assesses data losses at a lower data processing unit level.

The smallest accessed unit of a storage device is a *sector* in Hard-Disk Drives (HDDs), a *page* in flash-based Solid-State Drives (SSDs), and a *data set* in Linear Tape-Open (LTO is the trademark of HP, IBM, and Quantum in the Unites States and other countries) tape systems [8]. A sector has a typical size of 512 bytes or 4 KB, a page has a size that ranges from 4 KB to 16 KB, and a data set currently has a size of 5 MB or more. Erasure-coding redundancy schemes are implemented by treating the units that contain user data as symbols and complementing them with parity symbols (units) to form codewords. In the case of HDDs and SSDs, one or more units are allocated to an entity and the last unit may be partially filled. Depending on the file system employed, the remaining space of a partially-filled unit may or may not be used to store the contents of another entity. Therefore, user data may or may not be stored in an aligned fashion with units (symbols), which in turn implies that entities may or may not be aligned with codewords. The case where entities are aligned with codewords was considered by the reliability model presented in [7]. By contrast, in the case of tape, user data is written sequentially such that a unit may contain data of multiple entities. Therefore, user data and entities are not aligned with symbols and codewords, respectively. Moreover, the reliability model presented in [7] assumed that entities have a fixed size, whereas in practice they have variable sizes. It turns out that the MTTDL metric does not depend on the placement and size of the entities, but the EAFEL metric does. More specifically, EAFEL depends on the number of codewords that stored entities span. Furthermore, the EAFEL metric reflects the fraction of lost user data only when entities have a fixed size. To evaluate system durability in the case of variable-size entities, in this article we introduce the Expected Annual Fraction of Effective Data Loss (EAFEDL) reliability metric, that is, the fraction of stored user data that is expected to be lost by the system annually at the entity level.

The key contributions of this article are the following. The reliability model presented in [7] for the assessment of the EAFEL metric is enhanced in two ways. First, entities are considered to be stored such that they are not aligned with codeword boundaries. Second, the size of entities is considered to be variable. The objective of this article is to assess system reliability by deriving the distribution of the number of codewords that entities span. We address the following question. Does this distribution only depend on the statistics of the entities stored, that is, on their size and frequency of occurrence, or does it also depend on their placement? In the present work, we shed light on this issue by investigating the cases of deterministic and of random entity placement. The

distribution of the number of codewords that entities span is obtained analytically as a function of the size of the entities and the frequency of their occurrence. We also establish that for certain deterministic placements of variable-size entities, this distribution also depends on the actual entity placement.

The general non-Markovian methodology that was applied in prior work to assess the EAFDL and EAFEL metrics for erasure-coding redundancy schemes and for the clustered, declustered, and symmetric data placement schemes, was extended to derive analytically the EAFEL and the new EAFEDL reliability metrics for the case of variable-size entities [1]. It was demonstrated how the erasure-coding capability as well as the entity and symbol sizes affect system reliability in the entire range of bit error rates. In this article, we extend our previous work by deriving MTTDL for the case of lazy rebuilds and in the presence of correlated latent symbol errors. We also evaluate the EAFEL and EAFEDL reliability metrics for some real-world erasure coding schemes employed by enterprises. The model developed provides useful insights into the benefits of the erasure coding schemes and yields results for the entire parameter space, which allows a better understanding of the design tradeoffs.

The remainder of the article is organized as follows. Section II reviews prior relevant work and analytical models presented in the literature for assessing the effect of latent errors on the reliability of erasure-coded systems. Section III describes the storage system model and the corresponding parameters considered. In Section IV, the distribution of the number of codewords that entities span is derived analytically as a function of the entity size distribution when entities are not aligned with symbols and when entity sizes are either fixed or variable. In Section V, the MTTDL metric is derived analytically for the case of lazy rebuilds and correlated latent symbol errors. Also, the EAFEL and EAFEDL metrics are derived analytically for the case of random placement of variable-size entities. Section VI presents numerical results demonstrating the effect of the erasure-coding capability and of the entity sizes on system reliability, as well as the adverse effect of an increased symbol size. The reliability of real-world erasure coding schemes employed by enterprises to protect their stored data is assessed in Section VII. Finally, we conclude in Section VIII.

II. Related Work

Analytical reliability expressions for MTTDL that take into account the effect of latent errors have been obtained predominately using Markovian models, which assume that component failure and rebuild times are independent and exponentially distributed [9][10][11][12]. The effect of latent errors on MTTDL and EAFDL of erasure-coded storage systems for the realistic case of non-exponential failure and rebuild time distributions was assessed in [4][5].

Disk scrubbing has been used to mitigate the adverse effect of latent errors on system reliability [9][13][14][15]. The scrubbing process identifies latent errors at an early stage and attempts to correct them before disk failures occur. This in effect reduces the probability of encountering a latent error during the rebuild process. The resulting latent-error probability was derived in [9] as a function of the scrubbing and workload parameters. Subsequently, it was shown that the reliability level achieved when scrubbing is used can be obtained from the reliability level of a system that does not use scrubbing by adjusting the probability of encountering a latent error accordingly. The methodology presented in [9] for deriving the adjusted latent error probability when scrubbing is employed is also applicable for assessing the efficiency of other scrubbing schemes, such as the adaptive scrubbing schemes proposed in [14][15]. Moreover, this methodology can also be applied in conjunction with the reliability results presented in this article to assess the reliability of erasure-coded systems when scrubbing is used.

The efficiency of applying erasure coding in storage systems that employ solid state disks (SSDs) was studied in [16]. It was demonstrated that the reliability improvement achieved by erasure coding is in general greater than the reliability degradation induced. Also, the reliability of SSD arrays using a real-system implementation of conventional and emerging erasure codes was investigated in [17] using realistic storage traces.

A simulation analysis of reliability aspects of erasure-coded data centers was presented in [18]. Various configurations were considered and it was shown that erasure codes and redundancy placement affect system reliability. In [19] it was recognized that it is hard to get statistically meaningful experimental reliability results using prototypes, because this would require a large number of machines to run for years. This underscores the usefulness of the analytical reliability results derived in this article.

III. Storage System Model

The reliability of erasure-coded storage systems was assessed in [7] based on a model that considers codeword rebuilds for reconstructing lost symbols and assess system reliability when entities (files, objects, blocks) are lost. Maximum Distance Separable (MDS) erasure codes (m, l) that map l user-data symbols to codewords of m symbols are employed. They have the property that any subset containing l of the m codeword symbols can be used to reconstruct (recover) a codeword. The MTTDL and EAFEL reliability metrics were derived analytically for systems that employ a lazy rebuild scheme.

The corresponding storage efficiency s_{eff} and amount U of user data stored in the system is

$$s_{\text{eff}} = l/m \quad \text{and} \quad U = s_{\text{eff}}\, n\, c \;=\; l\, n\, c/m\;, \qquad (1)$$

where n is the number of storage devices in the system and c is the amount of data stored on each device. The storage space of devices is partitioned into units (symbols) of a fixed size s, such that the number C of symbols stored in a device is

$$C = c/s\;. \qquad (2)$$

Our notation is summarized in Table I. The parameters are divided according to whether they are independent or derived and are listed in the upper and lower part of the table, respectively.

To minimize the risk of permanent data loss, the m symbols of each of codeword are spread and stored in m devices. This

TABLE I. NOTATION OF SYSTEM PARAMETERS

Parameter	Definition
n	number of storage devices
c	amount of data stored on each device
l	number of user-data symbols per codeword ($l \geq 1$)
m	total number of symbols per codeword ($m > l$)
(m, l)	MDS-code structure
e_s	entity size
s	symbol (sector or data set) size
k	spread factor of the data placement scheme, or group size (number of devices in a group) ($m \leq k \leq n$)
b	average reserved rebuild bandwidth per device
B_{max}	upper limitation of the average network rebuild bandwidth
X	time required to read (or write) an amount c of data at an average rate b from (or to) a device
$F_X(.)$	cumulative distribution function of X
$F_\lambda(.)$	cumulative distribution function of device lifetimes
P_{b}	probability of an unrecoverable bit error
s_{eff}	storage efficiency of redundancy scheme ($s_{\text{eff}} = l/m$)
U	amount of user data stored in the system ($U = s_{\text{eff}}\, n\, c$)
$\tilde{r}$	MDS-code distance: minimum number of codeword symbols lost that lead to permanent data loss ($\tilde{r} = m - l + 1$ and $2 \leq \tilde{r} \leq m$)
C	number of symbols stored in a device ($C = c/s$)
μ^{-1}	mean time to read (or write) an amount c of data at an average rate b from (or to) a device ($\mu^{-1} = E(X) = c/b$)
λ^{-1}	mean time to failure of a storage device ($\lambda^{-1} = \int_0^\infty [1 - F_\lambda(t)]dt$)
P_s	probability of an unrecoverable sector (symbol) error
s_s	shard size ($s_s = e_s/l$)
J	shard size measured in symbol-size units ($J = s_s/s = e_s/(l\, s)$)
Y	number of lost entities during rebuild
$\breve{Q}$	amount of lost user data during rebuild

way, the system can tolerate any $\tilde{r} - 1$ device failures, but $\tilde{r}$ device failures may lead to data loss, with

$$\tilde{r} = m - l + 1\,, \quad 1 \leq l < m \quad \text{and} \quad 2 \leq \tilde{r} \leq m\,. \tag{3}$$

Examples of MDS erasure codes are the following:

Replication: A replication-based system with a replication factor r can tolerate any loss of up to $r - 1$ copies of some data, such that $l = 1$, $m = r$ and $\tilde{r} = r$. Also, its storage efficiency is equal to $s_{\text{eff}}^{\text{replication}} = 1/r$. The mirroring scheme is the special case where $r = 2$. The corresponding storage efficiency of only 50% can be improved by employing erasure codes.

RAID-5: A RAID-5 array comprised of N devices uses an $(N, N-1)$ MDS code, such that $l = N - 1$, $m = N$ and $\tilde{r} = 2$. It can therefore tolerate the loss of up to one device, and its storage efficiency is equal to $s_{\text{eff}}^{\text{RAID-5}} = (N-1)/N$.

RAID-6: A RAID-6 array comprised of N devices uses an $(N, N-2)$ MDS code, such that $l = N - 2$, $m = N$ and $\tilde{r} = 3$. It can therefore tolerate a loss of up to two devices, and its storage efficiency is equal to $s_{\text{eff}}^{\text{RAID-6}} = (N-2)/N$.

In terms of encoding operations, MDS erasure codes are either bitwise exclusive-OR (XOR) or non-XOR. The computation complexity of the non-XOR-based codes, such as Reed–Solomon, is much higher than that of the XOR-based ones. Also, in the context of storage, Reed-Solomon codes are preferable to Turbo codes owing to their simpler implementation and the fact that they are more suitable in environments where bit error rates are low, and errors occur in bursts.

Two different ways (A and B) for storing user data on devices were shown in Figure 1 of [7]. According to way A, user data contained in entities is divided into chunks with the contents of a chunk stored on different devices, whereas according to way B, user data contained in entities is divided into *shards* with the contents of a shard stored on the same device. More specifically, according to way B, user data contained in entities is divided into l shards with each one being stored on a different device, as shown in Figure 1(a). Entities were assumed to have a fixed size e_s with the corresponding shard size s_s then obtained by $s_s = e_s/l$.

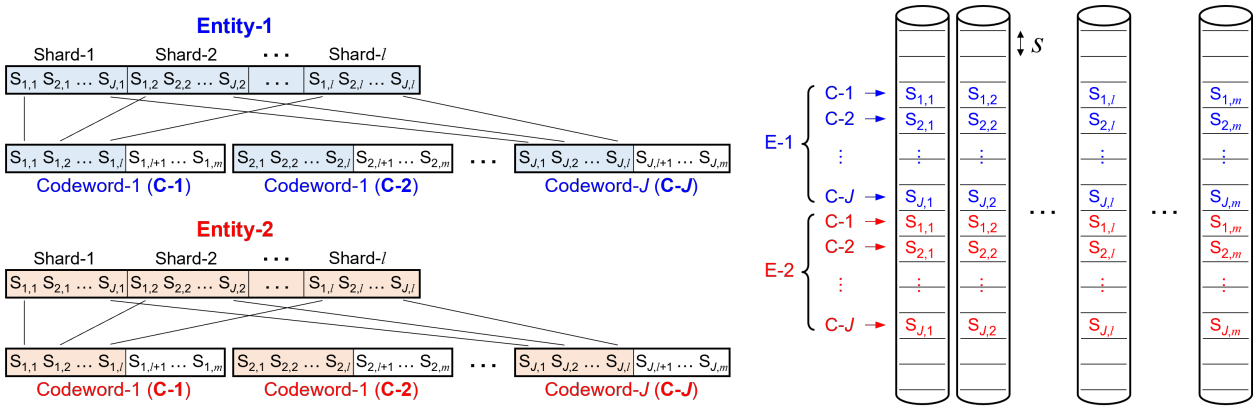
The storage space of devices is partitioned into units (symbols) of a fixed size s and complemented with parity symbols to form codewords. Each shard was assumed to be stored in an integer number of J symbols that is determined by

$$J = \frac{s_s}{s} = \frac{e_s}{l\, s}\,. \tag{4}$$

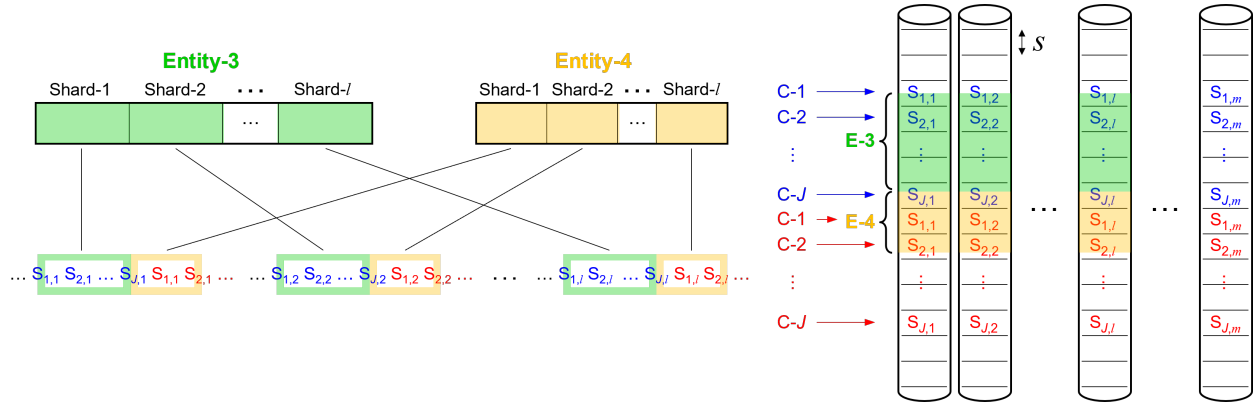
Consequently, the contents of each entity, such as Entity-1 and Entity-2, are stored in $J\, l$ user-data symbols with these symbols being stored in an integer number of J codewords. These codewords also contain $J\,(m - l)$ parity symbols for a total number of $J\, m$ symbols per entity, as shown in Figure 1(a). Note that $S_{j,i}$ denotes the ith symbol of the jth codeword. Thus, $S_{1,2}$, which is the second symbol of codeword C-1, is the first symbol of the second shard. Successive symbols of a shard are stored on the same device. To minimize the risk of permanent data loss, the m symbols of each of the J codewords are spread and stored successively in a set of m devices.

The model in [7] considered shards that have a fixed size of J symbols and are stored aligned with the symbol boundaries, which are indicated by the horizontal black lines in Figure 1(a). However, in practice user entities, and in turn shards, do not have a fixed size and, in the case of tape, are not necessarily aligned with symbols, because, as discussed in Section I, entity data is stored in a way that is agnostic to symbol boundaries. This is demonstrated in Figure 1(b) that shows two entities of two different sizes, Entity-3 and Entity-4, and the way they are stored on l devices of the system. For instance, Shard 1 of Entity-3 spans J symbols, i.e., the blue symbols $S_{1,1}$, $S_{2,1}$, $\cdots$, $S_{J,1}$, with its data partially occupying the first and last symbol, $S_{1,1}$ and $S_{J,1}$, respectively. Subsequently, Shard 1 of Entity-4 spans three symbols, namely, the blue symbol $S_{J,1}$ and the two red symbols $S_{1,1}$ and $S_{2,1}$, with its data partially occupying the first and the last symbol, that is, the blue $S_{J,1}$ and the red $S_{2,1}$ symbol. Thus, symbol $S_{J,1}$ contains data from both these entities. More generally, depending on the entity and symbol sizes, a symbol may contain data from multiple entities. Clearly, shard and entity sizes do not necessarily correspond to an integer number of symbols, which implies that the size J of a shard, expressed in number of symbols by (4), is in general a real number, which is less than 1 when the shard size is less than the symbol size. Codewords are formed by combining symbols containing user-data to generate and store parity symbols, as shown in Figure 1(b), regardless of the entities involved.

As pointed out in [7], the MTTDL metric does not depend on the entity size. This is due to the fact that the degree to which permanent data losses occur depends on the capability of the erasure-coding redundancy scheme employed and the resulting codeword formation, which in turn is agnostic to the entity placement and size characteristics. Note that an entity is lost if any of the codewords that it spans is permanently



(a) Symbol-aligned shards of integer size



(b) Non-symbol-aligned shards of arbitrary size

Figure 1. Data placement of entities and formation of codewords.

lost. Consequently, the EAFEL and EAFEDL metrics, which consider data loss at the entity level, depend on the number of codewords that entities span. The corresponding derivation is performed in Section IV.

The reliability of storage systems degrades by the presence of unrecoverable or latent errors. According to the specifications of enterprise quality HDDs, the unrecoverable bit-error probability P_{b} is equal to 10^{-15}. In practice, however, P_{b} can be orders of magnitude higher, reaching $P_{\mathrm{b}} \approx 10^{-12}$ [5]. On the other hand, according to Figure 13 in [20], tapes are more reliable than HDDs with a Bit Error Rate (BER) in the range of 10^{-22} to 10^{-19}. Assuming that bit errors occur independently over successive bits, the unrecoverable symbol error probability P_s is determined by

$$P_s = 1 - (1 - P_{\mathrm{b}})^s \,, \tag{5}$$

with the symbol size s expressed in bits. For a symbol size of 512 bytes, the equivalent unrecoverable sector error probability is $P_s \approx P_{\mathrm{b}} \times 512 \times 8$, which is 4.096×10^{-12} and 4.096×10^{-9} for $P_{\mathrm{b}} \approx 10^{-15}$ and 10^{-12}, respectively. Moreover, latent errors are found to exhibit spatial locality and they occur in bursts of B contiguous symbol errors. The degree to which symbol errors are correlated is captured by the factor f_{cor} whose value is determined by [5, Eq. (29)]

$$f_{\mathrm{cor}} = \begin{cases} 1 \,, & \text{for independent symbol errors} \\ \frac{1}{\bar{B}} \,, & \text{for correlated symbol errors} \,, \end{cases} \tag{6}$$

where $\bar{B}$ denotes the average length (in number of symbols) of bursts of latent symbol errors. Thus, $f_{\mathrm{cor}} \geq 1$.

IV. Codewords Spanned by Entities

Here, we obtain the distribution of the number of codewords, K, that entities span, which also represents the number of symbols that shards span. We proceed by considering the cases of fixed- and variable-size entities (shards).

A. Fixed-Size Entities

Let us consider fixed-size entities, which in turn result in fixed-size shards, such that J is fixed. Owing to periodicity, it suffices to study the process within a window of $S = J \times 10^k$ symbols, where k represents the number of decimal digits of J. This window corresponds in a symbol interval $[\epsilon, S + \epsilon]$ where ϵ is the starting position of the first shard within the first symbol, such that $0 < \epsilon < 1$. This interval contains S symbol boundaries and stores 10^k shards. For example, for $J = 4.287$, we have $k = 3$, and it suffices to consider the process in a window of $S = 4.287 \times 10^3 = 4,287$ symbols that store 1000 shards.

Let us now consider the example shown in Figure 2 whereby the shard size is 2.3. In this case, it holds that $k = 1$ and therefore it suffices to consider the process within a window of $S = 2.3 \times 10^1 = 23$ symbols that store 10 shards depicted between the black circles with the symbol boundaries

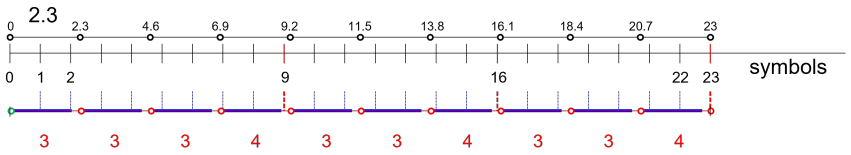


Figure 2. Number of symbols that shards span. Fixed-size shards of size $J = 2.3$ symbols.

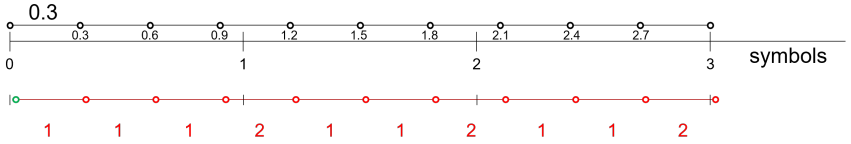


Figure 3. Number of symbols that shards span. Fixed-size shards of size $J = 0.3$ symbols.

indicated by the black vertical lines and with the first shard aligned with the first symbol. However, given that in practice shards are not aligned with symbols, their actual placement is indicated between the red circles, with the first shard starting at position ϵ, as indicated by the green circle. Figure 2 shows the case where $\epsilon = 0^+$.

Owing to periodicity, it suffices to study the process in the symbol interval $[\epsilon, 23 + \epsilon]$. The red integers indicate the number of symbols spanned by the successive shards. We note that 7 shards span 3 symbol and the remaining 3 shards span 4 symbols. Note that this holds for any $\epsilon \in (0, 1)$. Therefore, the probability density function (pdf) $\{p_j\}$ of the number of symbols K that an arbitrary shard spans is

$$P(K = i) = p_i = \begin{cases} 0.7\,, & \text{for } \; i = 3 \\ 0.3\,, & \text{for } \; i = 4\,. \end{cases} \tag{7}$$

Returning to the general case, we note that each shard can be decomposed into two components. The size of the first components, as indicated by the horizontal blue lines shown in Figure 2, corresponds to the number of symbols determined by the integer part of the shard size J, which is $\lfloor J \rfloor$ symbols. In the example considered, the integer part is 2. The size of the second components, as indicated by the horizontal red lines shown in Figure 2, corresponds to the fractional part, which is $J - \lfloor J \rfloor$ symbols. In the example considered, the fractional part is 0.3. Clearly, to each of the first (blue) components correspond $\lfloor J \rfloor$ symbol boundaries, which implies that each shard spans at least $\lfloor J \rfloor + 1$ symbols. In the example considered, to each of the first (blue) components correspond 2 symbol boundaries, as indicated by the blue vertical dotted lines, and, consequently, each shard spans at least 3 symbols.

As there are 10^k first components, one for each shard, the number of the corresponding symbol boundaries is $\lfloor J \rfloor \times 10^k$, which, in the example considered, is $2 \times 10^1 = 20$, as indicated by the blue vertical dotted lines. Consequently, there are $S - \lfloor J \rfloor \times 10^k = (J - \lfloor J \rfloor) \times 10^k$ additional symbol boundaries that correspond to $(J - \lfloor J \rfloor) \times 10^k$ out of the 10^k second components. In the example considered, there are $23 - 20 = 3$ additional symbol boundaries, as indicated by the red vertical dotted lines at positions 9, 16, and 23, that correspond to 3 out of the 10 red components. Consequently, these 3 components are associated with 3 shards, each of which spans one additional symbol for a total of 4 symbols. In general, each of the corresponding $(J - \lfloor J \rfloor) \times 10^k$ shards spans one additional symbol for a total of $\lfloor J \rfloor + 2$ symbols. Therefore, the percent of shards that span $\lfloor J \rfloor + 2$ symbols is $(J - \lfloor J \rfloor) \times 10^k / 10^k$ which is equal to $J - \lfloor J \rfloor$, that is, the fractional part of J denoted by $fr(J)$. Consequently, for any ϵ $(0 < \epsilon < 1)$, it holds that

$$P(K = i) = p_i = \begin{cases} 1 - fr(J)\,, & \text{for } \; i = \lfloor J \rfloor + 1 \\ fr(J)\,, & \text{for } \; i = \lfloor J \rfloor + 2 \\ 0\,, & \text{otherwise}\,, \end{cases} \tag{8}$$

where $fr(x)$ denotes the fractional part of the real number x,

$$fr(x) \triangleq x - \lfloor x \rfloor\,, \quad \forall\, x \in \mathcal{R}\,. \tag{9}$$

Let us also consider the case where $J < 1$ and the example shown in Figure 3 whereby the shard size is 0.3. Let us consider the first 10 shards indicated between the black circles with the first shard aligned with the first symbol. However, given that in practice shards are not aligned with symbols, their actual placement is indicated between the red circles, with the first shard starting at position ϵ, as indicated by the green circle. Owing to periodicity, it suffices to study the process in the symbol interval $[\epsilon, 3 + \epsilon]$. The red integers indicate the number of symbols spanned by the successive shards. We note that 7 shards span 1 symbol and the remaining 3 shards span 2 symbols and this holds for any $\epsilon \in (0, 1)$. Therefore, the pdf $\{p_j\}$ of the number of codewords (symbols) K that an arbitrary entity (shard) spans is

$$P(K = i) = p_i = \begin{cases} 0.7\,, & \text{for } \; i = 1 \\ 0.3\,, & \text{for } \; i = 2\,, \end{cases} \tag{10}$$

which is also the result determined by (8).

Next, we consider the case where the shard size is 2.7 symbols, as shown in Figure 4. Owing to periodicity, it suffices to study the process in the symbol interval $[\epsilon, 27 + \epsilon]$. The red integers indicate the number of symbols spanned by the successive shards. We note that 7 shards span 4 symbol and the remaining 3 shards span 3 symbols. According to (8), the pdf $\{p_j\}$ of the number of codewords (symbols) K that an arbitrary entity (shard) spans is

$$P(K = i) = p_i = \begin{cases} 0.3\,, & \text{for } \; i = 3 \\ 0.7\,, & \text{for } \; i = 4\,, \end{cases} \tag{11}$$

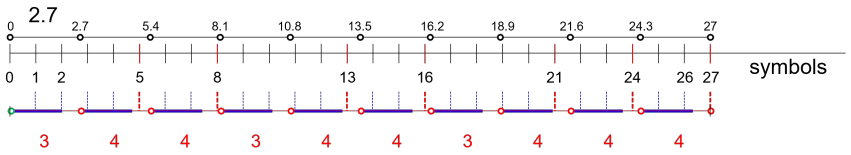


Figure 4. Number of symbols that shards span. Fixed-size shards of size $J = 2.7$ symbols.

which is also the result determined by (8).

B. Variable-Size Entities

We proceed to relax the assumption that all entities have the same size, by considering entities of E_s different sizes, $e_{s,1}, e_{s,2}, \cdots, e_{s,E_s}$. Without loss of generality, we assume that $e_{s,1} < e_{s,2} < \cdots < e_{s,E_s}$. Subsequently, let $\{v_j\}$ denote the corresponding pdf of the entity size, that is,

$$v_j \triangleq P(e_s = e_{s,j}) , \quad \text{for } j = 1, 2, \ldots, E_s , \tag{12}$$

such that the average entity size $E(e_s)$ is determined by

$$E(e_s) = \sum_{j=1}^{E_s} e_{s,j}\, v_j . \tag{13}$$

From (4), it follows that the shard size J_j corresponding to entity $e_{s,j}$ is determined by

$$J_j = \frac{e_{s,j}}{l\, s} \quad \text{for } j = 1, 2, \ldots, E_s . \tag{14}$$

Consequently, the pdf of the shard size J is determined by

$$P(J = J_j) = v_j , \quad \text{for } j = 1, 2, \ldots, E_s , \tag{15}$$

such that the average shard size $E(J)$ is determined by

$$E(J) = \sum_{j=1}^{E_s} J_j\, v_j \overset{(13)(14)}{=} \frac{E(e_s)}{l\, s} , \tag{16}$$

where the notation $\overset{(x)(y)}{=}$ implies that the final expression is derived using Equations (x) and (y).

The preceding discussion begs the following questions. Can the probability density function $\{p_j\}$ that was theoretically obtained in (8) for the case of a single fixed shard size be extended for the case of variable-size entities? Does it depend on the sequence according to which the variable-size entities are stored? Next, we address these critical questions. We shed light on these issues by considering the following cases regarding the placement and the way according to which the various shards are stored.

1) Segregated Shard Placement: According to this placement, shards of any given size are stored successively. One particular realization is to first store the shards of size J_1, followed by the shards of size J_2, and so on. For a large number of shards stored, from (8) and (15) we deduce that

$$P(K = i) = p_i = \begin{cases} [1 - fr(J_j)]\, v_j , & \text{for } i = \lfloor J_j \rfloor + 1 \\ fr(J_j)\, v_j , & \text{for } i = \lfloor J_j \rfloor + 2 \\ 0 , & \text{otherwise,} \end{cases}$$
$$\text{for } j = 1, 2, \ldots, E_s . \tag{17}$$

Let us consider the special case of a discrete bimodal distribution for the shard size, that is, $E_s = 2$, and let us assume that half of the shards have a size of 0.3 symbols and the remaining half of the shards have a size of 2.7 symbols. In this case we have $J_1 = 0.3$, $J_2 = 2.7$, and $v_1 = v_2 = 0.5$. For the particular realization where first the shards of size 0.3 are stored followed by the shards of size 2.7, (17) yields

$$P(K = i) = p_i = \begin{cases} 0.7 \times 0.5 = 0.35 , & \text{for } i = 1 \\ 0.3 \times 0.5 = 0.15 , & \text{for } i = 2 \\ 0.3 \times 0.5 = 0.15 , & \text{for } i = 3 \\ 0.7 \times 0.5 = 0.35 , & \text{for } i = 4 \\ 0 , & \text{otherwise.} \end{cases} \tag{18}$$

2) Alternating Shard Placement: According to this placement, shards of various sizes are stored interleaved by also considering the v_j values. One particular realization in the case where $v_j = 1/E_s$, for $j = 1, 2, \ldots, E_s$, is to first store a shard of size J_1, followed by a shard of size J_2, and so on. The first cycle is completed by storing a shard of size J_{E_s} and is followed by a second cycle that begins by storing a shard of size J_1.

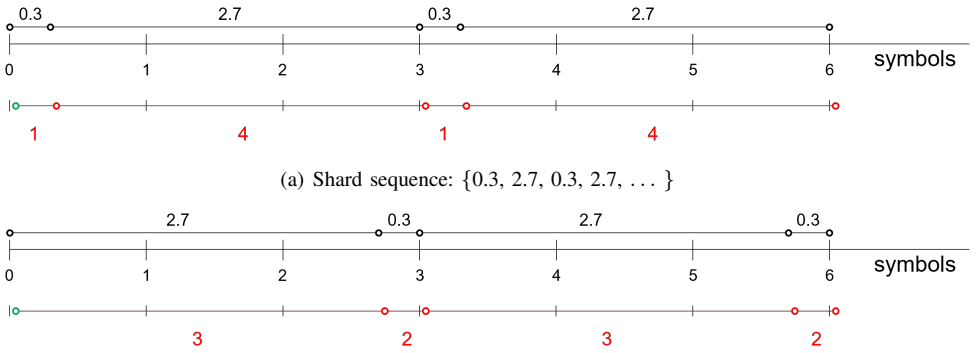
We proceed by investigating the special case considered in Section IV-B1 for the discrete bimodal distribution of the shard size, with the sizes of 0.3 and 2.7 symbols. The alternating placement of the shards corresponding to these two sizes lead to two possible sequence realizations, as shown in Figure 5.

The realization for the alternating sequence $\{0.3, 2.7, 0.3, 2.7, \ldots\}$ is depicted in Figure 5(a). Owing to periodicity, it suffices to study the process in the symbol interval $[\epsilon, 3+\epsilon]$. Figure 5(a) shows the case where $\epsilon = 0^+$. The red integers indicate the number of symbols spanned by the successive shards. We note that half of the shards span 1 symbol and the remaining half of the shards span 4 symbols and this holds for any $\epsilon \in (0, 0.7)$. Consequently, the pdf $\{p_j\}$ of the number of symbols K that an arbitrary shard spans is

$$P(K = i) = p_i = \begin{cases} 0.5 , & \text{for } i = 1 \\ 0.5 , & \text{for } i = 4 . \end{cases} \tag{19}$$

On the other hand, the realization for the alternating sequence $\{2.7, 0.3, 2.7, 0.3, \ldots\}$ is depicted in Figure 5(b). Owing to periodicity, it suffices to study the process in the symbol interval $[\delta, 3+\delta]$. Figure 5(b) shows the case where $\delta = 0^+$. In this case, half of the shards span 3 symbols and the remaining half of the shards span 2 symbols and this holds for any $\delta \in (0, 0.3)$. Consequently, the pdf $\{p_j\}$ of the number of symbols K that an arbitrary shard spans is

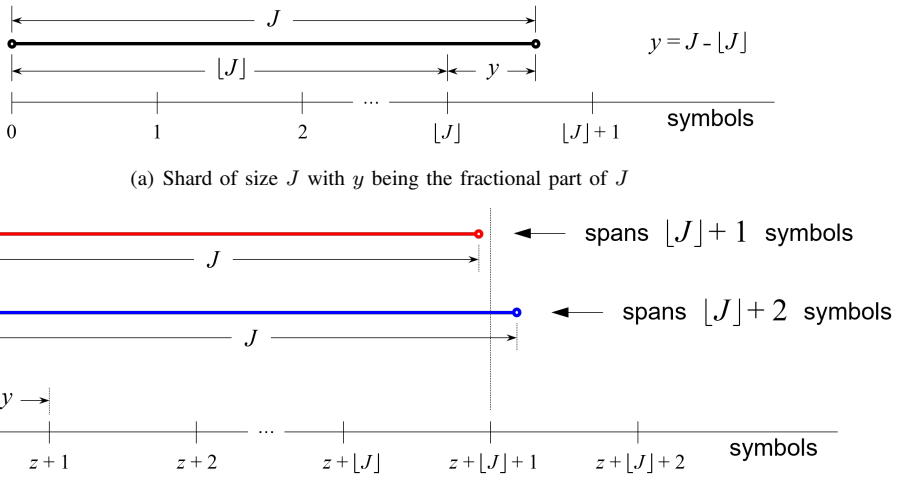
$$P(K = i) = p_i = \begin{cases} 0.5 , & \text{for } i = 2 \\ 0.5 , & \text{for } i = 3 . \end{cases} \tag{20}$$



(a) Shard sequence: {0.3, 2.7, 0.3, 2.7, . . . }

(b) Shard sequence: {2.7, 0.3, 2.7, 0.3, . . . }

Figure 5. Number of symbols spanned by shards. Alternating fixed-size shards of sizes 0.3 and 2.7 symbols, with $v_1 = v_2 = 0.5$.



(a) Shard of size J with y being the fractional part of J

(b) Random shard placement

Figure 6. Number of symbols that a randomly placed shard of size J spans.

Note that the pdf for $\delta \in (0.3, 1)$ is that determined by (19). Also, the pdf for $\epsilon \in (0.7, 1)$ is that determined by (20).

We now observe that the pdf determined by (20) is different from that determined by (19). Moreover, both of them, are different from that determined by (18) for the case of a segregated shard placement. Therefore, from the above, we deduce that the pdf $\{p_j\}$ of the number of symbols K that an arbitrary shard spans not only depends on the percentage of the various shard sizes in a sequence, as specified in (15), but also on their actual placement.

3) Random Shard Placement: According to this placement, the starting position ϵ $(0 < \epsilon < 1)$ of the first shard within the first symbol is uniformly distributed in $(0, 1)$. Successive shard sizes are assumed to be identically distributed, according to the distribution given in (15), but not necessarily independent. Note that this relaxes the assumption made in [1] of independent and identically distributed (i.i.d) successive shard sizes.

Let us consider a randomly chosen shard. Let also J denote its size, as shown in Figure 6(a), and y its fractional part, that is, $y = J - \lfloor J \rfloor$. Owing to the random placement of the first shard, the chosen shard, too, is randomly placed, such that it spans either $\lfloor J \rfloor + 1$ or $\lfloor J \rfloor + 2$ symbols, as depicted by the red and the blue shards shown in Figure 6(b), respectively. Let X denote the distance between the starting position of the shard and the left boundary z of the first symbol that the shard spans. Owing to the random placement of the shard, the random variable X is uniformly distributed between 0 and 1. Furthermore, when $X \leq 1 - y$, the shard spans $\lfloor J \rfloor + 1$ symbols whereas when $X > 1 - y$, the shard spans $\lfloor J \rfloor + 2$ symbols. Consequently, the probability that the shard spans $\lfloor J \rfloor + 1$ symbols is

$$P(K = \lfloor J \rfloor + 1) = \int_0^{1-y} dx = 1 - y \,, \tag{21}$$

which implies that the probability that the shard spans $\lfloor J \rfloor + 2$ symbols is

$$P(K = \lfloor J \rfloor + 2) = 1 - P(K = \lfloor J \rfloor + 1) \overset{(21)}{=} y \,. \tag{22}$$

Therefore, and given that $y = J - \lfloor J \rfloor = fr(J)$, it holds that

$$P(K = i) = p_i = \begin{cases} 1 - fr(J) \,, & \text{for } \; i = \lfloor J \rfloor + 1 \\ fr(J) \,, & \text{for } \; i = \lfloor J \rfloor + 2 \\ 0 \,, & \text{otherwise} \,. \end{cases} \tag{23}$$

From (23), and using (9), it follows that the mean number $E(K)$ of symbols that a shard of size J spans is

$$E(K) = (\lfloor J \rfloor + 1)P(K = \lfloor J \rfloor + 1) + (\lfloor J \rfloor + 2)P(K = \lfloor J \rfloor + 2) = (\lfloor J \rfloor + 1)\,[1 - fr(J)] + (\lfloor J \rfloor + 2)\,fr(J) = J + 1\,. \tag{24}$$

From (15), (23), and (24), it follows that the pdf and the average number of symbols K that an arbitrary shard spans are determined by

$$P(K = i) = p_i = \begin{cases} [1 - fr(J_j)]\,v_j\,, & \text{for } \; i = \lfloor J_j \rfloor + 1 \\ fr(J_j)\,v_j\,, & \text{for } \; i = \lfloor J_j \rfloor + 2 \\ 0\,, & \text{otherwise,} \end{cases} \quad \text{for } j = 1, 2, \ldots, E_s\,, \tag{25}$$

and

$$E(K) = \sum_{j=1}^{E_s} (\,J_j + 1)\,v_j = E(J) + 1\,. \tag{26}$$

Remark 1: For two different shard-size values, say $J_m \neq J_n$, for which it holds that $\lfloor J_m \rfloor = \lfloor J_n \rfloor = j$, the corresponding probabilities of the number of symbols K that these shards span are determined additively, that is, $P(K = j + 1) = [1 - fr(J_m)]\,v_m + [1 - fr(J_n)]\,v_m$ and $P(K = j + 2) = fr(J_m)\,v_m + fr(J_n)\,v_n$. Similarly, if $\lfloor J_m \rfloor + 1 = \lfloor J_n \rfloor = j$, then it holds that $P(K = j + 1) = fr(J_m)\,v_m + [1 - fr(J_n)]\,v_n$.

Remark 2: From (17) and (25), it follows that the pdfs of the number of symbols K that an arbitrary shard spans in the segregated and the random shard placement cases are the same. This is also the pdf that corresponds to the alternating shard placement case when the first shard is randomly placed. For the discrete bimodal distribution considered in Section IV-B2 for the alternating shards of sizes 0.3 and 2.7 symbols, when the first shard is randomly placed, that is, when the variables ϵ and δ are uniformly distributed in $(0, 1)$, combining (19) and (20) yields a pdf that is the same as that derived in (18) for the segregated shard placement case. Clearly, in the segregated and alternating shard placement cases successive shard sizes are dependent.

V. Derivation of MTTDL, EAFEL, and EAFEDL

The MTTDL, EAFEL, and EAFEDL reliability metrics are derived using the direct-path-approximation methodology presented in [2-6] and extend it to assess the effect of lazy rebuilds [21] in the presence of correlated symbol errors [5].

At any point in time, the system is in one of two modes: non-rebuild or rebuild mode. Note that part of the non-rebuild mode is the normal mode of operation where all devices are operational and all data in the system has the original amount of redundancy. Upon device failures, a rebuild process attempts to restore the lost data, which eventually leads the system either to a Data Loss (DL) with probability $P_{\rm DL}$ or back to the original normal mode by restoring initial redundancy, with probability $1 - P_{\rm DL}$. The MTTDL metric is then obtained by [6, Eq. (5)]:

$$\text{MTTDL} \approx \frac{E(T)}{P_{\rm DL}}\,, \tag{27}$$

where $P_{\rm DL}$ is determined by (49) and $E(T)$ denotes the expected duration, expressed in years, of a typical interval of normal operation until the rebuild process of failed devices is triggered, which is determined by Eq. (12) of [21] as follows:

$$E(T) = \left(\sum_{u=0}^{d} \frac{1}{\tilde{n}_u} \right) \Big/ \lambda\,, \quad \text{where } \tilde{n}_0 \triangleq n\,, \tag{28}$$

where $1/\lambda$ is the mean time to failure of a device and $\tilde{n}_u$ are determined by (35), (38), or (41), depending on the data placement scheme.

The EAFEL metric is obtained by Eq. (16) of [7] as follows:

$$\text{EAFEL} \approx \frac{E(Y)}{E(T) \cdot N_E}\,, \tag{29}$$

that is, as the ratio of the expected number $E(Y)$ of lost entities, normalized to the number N_E of entities in the system, to the expected duration $E(T)$ expressed in years. The number N_E of entities in the system is determined by

$$N_E \approx \frac{U}{E(e_s)} \overset{(1)}{=} \frac{n}{m} \cdot \frac{l\,c}{E(e_s)} \overset{(16)}{=} \frac{n}{m} \cdot \frac{c}{E(J)\,s}\,, \tag{30}$$

and $E(T)$ and $E(Y)$ are determined by (28) and (64).

Analogous to Eq. (9) of [6], the EAFEDL is obtained as the ratio of the expected amount $E(\breve{Q})$ of lost user data at the entity level, normalized to the amount U of user data, to the expected duration of $E(T)$ expressed in years:

$$\text{EAFEDL} \approx \frac{E(\breve{Q})}{E(T) \cdot U} \overset{(1)}{=} \frac{m\,E(\breve{Q})}{n\,l\,c\,E(T)}\,, \tag{31}$$

where $E(T)$ and $E(\breve{Q})$ are determined by (28) and (84).

A. Reliability Analysis

The EAFEL and EAFEDL are evaluated in parallel with MTTDL using the theoretical framework presented in [7]. The system is at exposure level u $(0 \leq u \leq \tilde{r})$ when there are codewords that have lost u symbols owing to device failures, but there are no codewords that have lost more symbols. These codewords are referred to as the *most-exposed* codewords. Transitions to higher exposure levels are caused by device failures, whereas transitions to lower ones are caused by successful rebuilds. We denote by C_u the number of most-exposed codewords upon entering exposure level u, $(u \geq 1)$. Upon the first device failure it holds that

$$C_1 = C\,, \tag{32}$$

where C is determined by (2).

A certain portion of the device bandwidth is reserved for read/write data recovery during the rebuild process, and the remaining bandwidth is used to serve user requests. Let b denote the actual average reserved rebuild bandwidth per device. Lost symbols are rebuilt in parallel using the rebuild bandwidth b available on each surviving device. The amount of data corresponding to the number C_u of symbols to be rebuilt at exposure level u is written at an average rate b_u $(\leq b)$ to selected device(s). For the time X required to read (or write) an amount c of data from (or to) a device it holds that

$$E(X) = c/b\,. \tag{33}$$

The results in this article hold for *highly reliable* storage devices, which satisfy the following condition [5][7]

$$\mu \int_0^\infty F_\lambda(t)[1 - F_X(t)]dt \ll 1, \quad \text{with} \quad \frac{\lambda}{\mu} \ll 1 \,. \tag{34}$$

This condition expresses the fact that the ratio of the mean time $1/\mu$ to read all contents of a device (which typically is on the order of tens of hours) to the mean time to failure of a device $1/\lambda$ (which is typically at least on the order of thousands of hours) is very small, and in particular the fact that it is very unlikely that a given device fails during a rebuild period.

At exposure level u, the number $\tilde{n}_u$ of devices whose failure causes an exposure level transition to level $u+1$ and the fraction V_u of the C_u most-exposed codewords that have symbols stored on any given such device depend on the codeword placement scheme. In particular, for the symmetric and declustered data placement, at each exposure level u, for $u = 1, \cdots, \tilde{r}-1$, it holds that [2][3]

$$\tilde{n}_u^{\text{sym}} = \; k - u \,, \qquad \text{for} \;\; u = 1, \ldots, \tilde{r} \tag{35}$$

$$b_u^{\text{sym}} = \frac{\min((k-u)\, b, B_{\max})}{l+1} \,, \quad \text{for} \;\; u = d+1, \ldots, \tilde{r} \tag{36}$$

$$V_u^{\text{sym}} = \; \frac{m-u}{k-u} \,, \qquad \text{for} \;\; u = 1, \ldots, \tilde{r} \,, \tag{37}$$

where $B_{\max}$ is the maximum network rebuild bandwidth.

The corresponding parameters $\tilde{n}_u^{\text{declus}}$, b_u^{declus}, and V_u^{declus} for the declustered placement are derived from (35), (36), and (37) by setting $k = n$ as follows:

$$\tilde{n}_u^{\text{declus}} = \; n - u \,, \qquad \text{for} \;\; u = 1, \ldots, \tilde{r} \tag{38}$$

$$b_u^{\text{declus}} = \frac{\min((n-u)\, b, B_{\max})}{l+1} \,, \quad \text{for} \;\; u = d+1, \ldots, \tilde{r} \tag{39}$$

$$V_u^{\text{declus}} = \; \frac{m-u}{n-u} \,, \qquad \text{for} \;\; u = 1, \ldots, \tilde{r} \,. \tag{40}$$

For the clustered placement, it holds that [2][3]

$$\tilde{n}_u^{\text{clus}} = \; m - u \,, \qquad \text{for} \;\; u = 1, \ldots, \tilde{r} \tag{41}$$

$$b_u^{\text{clus}} = \; \min(\, b \,, B_{\max}/l \,) \,, \qquad \text{for} \;\; u = d+1, \ldots, \tilde{r} \tag{42}$$

$$V_u^{\text{clus}} = \; 1 \,, \qquad \text{for} \;\; u = 1, \ldots, \tilde{r} \,. \tag{43}$$

Also, for the rebuild time R_u of the most-exposed codewords at exposure level u and for its fraction α_u still left when another device fails, causing the exposure level transition $u \to u+1$, it holds that [21, Eq. (49)]

$$R_{d+1} \approx \left(\prod_{j=1}^{d} V_j \right) \frac{b}{b_{d+1}} \, X \,, \tag{44}$$

with the convention that for any integer j and for any sequence δ_i , $\prod_{i=j}^{0} \delta_i \triangleq 1$.

For $u \leq d$, no rebuild is performed and therefore $\alpha_u = 1$. For $u > d$, α_u is approximately uniformly distributed in $(0,1)$ such that [21, Eq. (8)],

$$\alpha_u \;\approxeq\; \begin{cases} 1 \,, & \text{for} \;\; u = 1, \ldots, d \\ U(0,1) \,, & \text{for} \;\; u = d+1, \ldots, \tilde{r}-1 \,. \end{cases} \tag{45}$$

TABLE II. NOTATION OF RELIABILITY METRICS AT EXPOSURE LEVELS

Parameter	Definition
u	exposure level
P_u	probability of entering exposure level u
P_{UF_u}	probability of data loss due to unrecoverable symbol errors at exposure level u
P_{UF}	probability of data loss due to unrecoverable symbol errors
P_{DF}	probability of data loss due to $\tilde{r}$ successive device failures
P_{DL}	probability of data loss
q_u	probability that, at exposure level u, a codeword that has lost u symbols can be restored
$\hat{q}_u$	probability that, under instantaneous transitions from exposure level 1 to exposure level u, all of the C_u most-exposed codewords, which have lost u symbols, can be restored
$\tilde{q}_u$	probability that, at exposure level u, an arbitrary entity is lost
$\breve{q}_u$	expected amount of lost user data of an arbitrary entity at exposure level u
$\tilde{q}_{s,u}(x)$	the probability of loss, at exposure level u, of an entity whose shard size expressed in symbols is x

Furthermore, it holds that [21, Eq. (10)]

$$C_u \;\approx\; C \prod_{i=1}^{u-1} V_i \, \alpha_i \,, \quad \text{for} \;\; u = 1, \ldots, \tilde{r} \,, \tag{46}$$

The reliability metrics of interest are derived using the *direct path approximation*, which considers only transitions from lower to higher exposure levels [2-6]. This implies that each exposure level is entered only once. At any exposure level u $(u = d+1, \ldots, \tilde{r}-1)$, data loss may occur during rebuild owing to one or more unrecoverable failures, which is denoted by the transition $u \to \text{UF}$. Moreover, at exposure level $\tilde{r}-1$, data loss occurs owing to a subsequent device failure, which leads to the transition to exposure level $\tilde{r}$. Consequently, the direct paths that lead to data loss are the following:

$\overrightarrow{UF_u}$: the direct path of successive transitions $1 \to 2 \to \cdots \to u \to \text{UF}$, for $u = d+1, \ldots, \tilde{r}-1$, and

$\overrightarrow{\text{DF}}$: the direct path of successive transitions $1 \to 2 \to \cdots \to \tilde{r}-1 \to \tilde{r}$,

with corresponding probabilities P_{UF_u} and P_{DF}, respectively. The notation for the probabilities of the events that lead to data loss is summarized in Table II.

1) Data Loss: It holds that

$$P_{\text{UF}_u} = P_u \; P_{u \to \text{UF}} \,, \quad \text{for} \;\; u = d+1, \ldots, \tilde{r}-1 \,, \tag{47}$$

where P_u is the probability of entering exposure level u determined by [21, Eq. (17)]:

$$P_u \approx \frac{(\lambda\, c \prod_{j=1}^{d} V_j)^{u-d-1}}{(u-d-1)!} \, \frac{E(X^{u-d-1})}{[E(X)]^{u-d-1}} \prod_{i=d+1}^{u-1} \frac{\tilde{n}_i}{b_i} \, V_i^{u-1-i} \,, \tag{48}$$

and $P_{u \to \text{UF}}$ is the probability of encountering an unrecoverable failure during the rebuild process at this exposure level.

In [10], it was shown that P_{DL} is accurately approximated by the probability of all direct paths to data loss. Therefore,

$$P_{\text{DL}} \approx \; P_{\text{DF}} + \sum_{u=d+1}^{\tilde{r}-1} P_{\text{UF}_u} \,. \tag{49}$$

Approximate expressions for the probabilities of data loss P_{UF_u} and P_{DF} are subsequently obtained by the following proposition.

Proposition 1: For $u = d+1, \ldots, \tilde{r}-1$, it holds that

$$P_{\mathrm{UF}_u} \approx -\left(\lambda\, c \prod_{j=1}^{d} V_j\right)^{u-d-1} \frac{E(X^{u-d-1})}{[E(X)]^{u-d-1}} \left(\prod_{i=d+1}^{u-1} \frac{\tilde{n}_i}{b_i}\, V_i^{u-1-i}\right) \cdot \log(\hat{q}_u)^{-(u-d-1)} \left(\hat{q}_u - \sum_{i=0}^{u-d-1} \frac{\log(\hat{q}_u)^i}{i!}\right), \quad (50)$$

where

$$\hat{q}_u \triangleq q_u^{f_{\mathrm{cor}}\, C \prod_{j=1}^{u-1} V_j}, \quad (51)$$

$$q_u = 1 - \sum_{j=\tilde{r}-u}^{m-u} \binom{m-u}{j} P_s^j (1-P_s)^{m-u-j}, \quad (52)$$

$$P_{\mathrm{DF}} \approx \frac{(\lambda\, c \prod_{j=1}^{d} V_j)^{\tilde{r}-d-1}}{(\tilde{r}-d-1)!} \frac{E(X^{\tilde{r}-d-1})}{[E(X)]^{\tilde{r}-d-1}} \prod_{i=d+1}^{\tilde{r}-1} \frac{\tilde{n}_i}{b_i}\, V_i^{\tilde{r}-1-i}. \quad (53)$$

Proof: Immediate by combining Proposition 1 of [5] and Proposition 1 of [21], and by also taking into account the effect of correlated latent errors via the variable f_{cor}, which is determined by (6), as discussed in Appendix A. ■

Remark 3: For small values of P_s, and a according to Remark 1 of [5], it holds that

$$q_u \approx \begin{cases} 1 - \binom{m-u}{\tilde{r}-u} P_s^{\tilde{r}-u}, & \text{for } P_s \ll \binom{m-u}{\tilde{r}-u}^{-\frac{1}{\tilde{r}-u}} \\ 0\,, & \text{for } P_s \gg \binom{m-u}{\tilde{r}-u}^{-\frac{1}{\tilde{r}-u}}, \end{cases} \quad (54)$$

$$\hat{q}_u \approx \begin{cases} 1 - Z_u\, P_s^{\tilde{r}-u}, & \text{for } P_s \ll P^*_{s,u} \\ 0\,, & \text{for } P_s \gg P^*_{s,u}, \end{cases} \quad (55)$$

where

$$Z_u \triangleq f_{\mathrm{cor}}\, C \left(\prod_{j=1}^{u-1} V_j\right) \binom{m-u}{\tilde{r}-u}, \quad (56)$$

and $P^*_{s,u} = Z_u^{-\frac{1}{\tilde{r}-u}}$.

Corollary 1: For $u = d+1, \ldots, \tilde{r}-1$, it holds that

$$P_{\mathrm{UF}_u} \approx \begin{cases} A_u\, P_s^{\tilde{r}-u}, & \text{for } P_s \ll P^{(\tilde{r})}_{\tilde{s},u} \\ P_u\,, & \text{for } P_s \gg P^{(\tilde{r})}_{\tilde{s},u}, \end{cases} \quad (57)$$

where

$$A_u \triangleq f_{\mathrm{cor}}\, C \binom{m-u}{\tilde{r}-u} (\lambda\, c)^{u-d-1} \frac{\left(\prod_{j=1}^{d} V_j\right)^{u-d}}{(u-d)!} \cdot \frac{E(X^{u-d-1})}{[E(X)]^{u-d-1}} \left(\prod_{i=d+1}^{u-1} \frac{\tilde{n}_i}{b_i}\, V_i^{u-i}\right), \quad (58)$$

P_u is determined by (48), and

$$\boxed{P^{(\tilde{r})}_{\tilde{s},u} \triangleq \left[\frac{u-d}{f_{\mathrm{cor}}\, C \binom{m-u}{\tilde{r}-u} \prod_{i=1}^{u-1} V_i}\right]^{\frac{1}{\tilde{r}-u}}}. \quad (59)$$

Proof: See Appendix A. ■

Remark 4: It follows from (59) that $P^{(\tilde{r})}_{\tilde{s},u}$ is dominated by the large value of C. Consequently, it holds that

$$0 = P^{(\tilde{r})}_{\tilde{s},\tilde{r}} < P^{(\tilde{r})}_{\tilde{s},\tilde{r}-1} < \cdots < P^{(\tilde{r})}_{\tilde{s},d+2} < P^{(\tilde{r})}_{\tilde{s},d+1}\,. \quad (60)$$

Remark 5: Note that P_{DL}, as a function of P_s, exhibits $\tilde{r}-d$ plateaus at levels P_u in the intervals $(P^{(\tilde{r})}_{\tilde{s},u}, P^{(\tilde{r})}_{\hat{s},u})$, for $u = d+1, \cdots, \tilde{r}$, respectively, where $P^{(\tilde{r})}_{\hat{s},d+1} \triangleq 1$ and $P^{(\tilde{r})}_{\hat{s},u}$ is determined by (59). Also, $[0, P^{(\tilde{r})}_{\hat{s},u})$ is the range of values of P_s for which it holds that $P_{\mathrm{UF}_{u-1}} \ll P_{\mathrm{UF}_u}$. It follows from approximation (57) that $P^{(\tilde{r})}_{\hat{s},u}$ satisfies equation $A_{u-1}\,(P^{(\tilde{r})}_{\hat{s},u})^{\,\tilde{r}-u+1} = P_u$, which, using (2) and (48), yields

$$\boxed{P^{(\tilde{r})}_{\hat{s},u} \triangleq \left[\frac{\lambda\, s\, E(X^{u-d-1})\, \tilde{n}_{u-1}}{f_{\mathrm{cor}} \binom{m-u+1}{\tilde{r}-u+1} E(X)\, E(X^{u-d-2})\, b_{u-1}}\right]^{\frac{1}{\tilde{r}-u+1}}}. \quad (61)$$

Note also that when $P^{(\tilde{r})}_{\tilde{s},u} > P^{(\tilde{r})}_{\hat{s},u}$, the interval $(P^{(\tilde{r})}_{\tilde{s},u}, P^{(\tilde{r})}_{\hat{s},u})$ as well as the corresponding plateau vanish.

Remark 6: From (61), and given that the term in the bracket is quite small, it follows that

$$P^{(\tilde{r})}_{\hat{s},\tilde{r}} < P^{(\tilde{r})}_{\hat{s},\tilde{r}-1} < \cdots < P^{(\tilde{r})}_{\hat{s},d+2} < P^{(\tilde{r})}_{\hat{s},d+1} = 1\,. \quad (62)$$

The methodology presented in [10] that considers the most probable path to data loss yields an approximate function for P_{DL}. This function is obtained analytically by Corollary 1, and Remarks 4, 5, and 6, and has the shape shown in a log-log plot in Figure 7 along with the plateaus and corresponding intervals.

Remark 7: The plateaus derived in the case where $d = 0$ are in agreement with those determined in [5].

Remark 8: According to Remarks 5 and 6, P_{DL} and, by virtue of (27), MTTDL is affected when

$$\boxed{P_s \gg P^{(\tilde{r})}_{\hat{s},\tilde{r}} \overset{(3)(61)}{=} \frac{\lambda\, s\, E(X^{\tilde{r}-d-1})\, \tilde{n}_{\tilde{r}-1}}{f_{\mathrm{cor}}\, l\, E(X)\, E(X^{\tilde{r}-d-2})\, b_{\tilde{r}-1}}}. \quad (63)$$

2) Entity Loss: We proceed to derive the number of lost entities during rebuild. Let Y be the number of lost entities. Let also Y_{DF} and Y_{UF_u} denote the number of lost entities associated with the direct paths $\overrightarrow{\mathrm{DF}}$ and $\overrightarrow{UF_u}$, respectively. Then, it holds that [7, Eqs. (37), (38), (41)]

$$E(Y) \approx E(Y_{\mathrm{DF}}) + \sum_{u=d+1}^{\tilde{r}-1} E(Y_{\mathrm{UF}_u}) \approx E(Y_{\mathrm{DF}}) + E(Y_{\mathrm{UF}})\,, \quad (64)$$

where Y_{UF} denotes the number of lost entities due to unrecoverable failures with its mean given by

$$E(Y_{\mathrm{UF}}) \approx \sum_{u=d+1}^{\tilde{r}-1} E(Y_{\mathrm{UF}_u})\,. \quad (65)$$

Proposition 2: For $u = d+1, \ldots, \tilde{r}-1$, it holds that

$$E(Y_{\mathrm{UF}_u}) \approx \frac{C}{E(J)} \frac{P_u}{u-d} \left(\prod_{j=1}^{u-1} V_j\right) \tilde{q}_u\,, \quad (66)$$

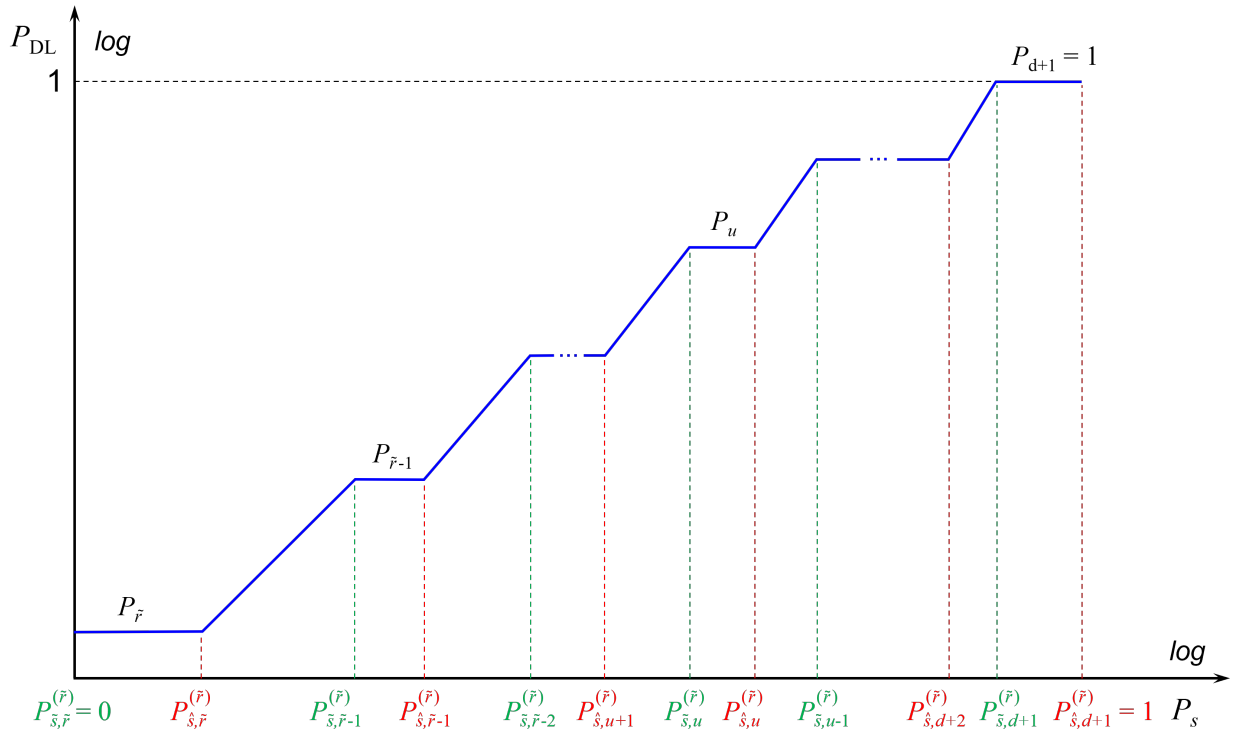


Figure 7. Approximate P_{DL} vs. P_s considering the most probable path to data loss.

where $\tilde{q}_u$, which denotes the probability that an arbitrary entity is lost, is determined by

$$\tilde{q}_u = \sum_{j=1}^{E_s} \tilde{q}_{s,u}\left(\frac{e_{s,j}}{l\,s}\right) v_j\,, \quad \text{for } u = d+1,\ldots,\tilde{r}\,, \tag{67}$$

with the probability $\tilde{q}_{s,u}(x)$ of loss, at exposure level u, of an entity whose shard size expressed in symbols is x, determined by

$$\tilde{q}_{s,u}(x) \triangleq 1 - [1 - fr(x)]\, q_u^{f_{\text{cor}}(\lfloor x \rfloor+1)} - fr(x)\, q_u^{f_{\text{cor}}(\lfloor x \rfloor+2)}\,, \tag{68}$$

and the probability q_u that a codeword that has lost u symbols can be restored, determined by (52).

It also holds that

$$E(Y_{\text{DF}}) \approx \frac{C}{E(J)}\,\frac{P_{\text{DF}}}{\tilde{r}-d}\prod_{j=1}^{\tilde{r}-1} V_j\,, \tag{69}$$

where C is determined by (2), P_s is determined by (5), $fr(x)$ is determined by (9), $E(J)$ is determined by (16). Also, the probability P_u of entering exposure level u is determined by (48).

Proof: Equation (66) is obtained in Appendix B. Equation (69) is obtained from (66) by setting $u = \tilde{r}$ and recognizing that $q_{\tilde{r}} = 0$, $\tilde{q}_{s,\tilde{r}}(x) = 1$, $\forall\, x \in \mathcal{R}$, $\tilde{q}_{\tilde{r}} = 1$, and $P_{\tilde{r}} = P_{\text{DF}}$. ■

Remark 9: For $u = d+1,\ldots,\tilde{r}-1$ and for small values of P_s, it follows from (68) and (54) that

$$\tilde{q}_{s,u}(x) \approx \begin{cases} f_{\text{cor}}\,(x+1)\binom{m-u}{\tilde{r}-u} P_s^{\tilde{r}-u}, & \text{for } P_s \ll P^{\#}_{s,u}(x) \\ 1\,, & \text{for } P_s \gg P^{\#}_{s,u}(x)\,, \end{cases} \tag{70}$$

where $P^{\#}_{s,u}(x)$ is obtained from the approximation (70), $\tilde{q}_{s,u}(x) \approx f_{\text{cor}}\,(x+1)\binom{m-u}{\tilde{r}-u} P^{\#}_{s,u}(x)^{\tilde{r}-u} = 1$, as follows: $P^{\#}_{s,u}(x) \triangleq \left[f_{\text{cor}}\,(x+1)\binom{m-u}{\tilde{r}-u}\right]^{-\frac{1}{\tilde{r}-u}}$. Also, for $u = \tilde{r}$, and given that $q_{\tilde{r}} = 0$, it follows from (68) that

$$\tilde{q}_{s,\tilde{r}}(x) = 1\,, \quad \forall\, x \in \mathcal{R}\,. \tag{71}$$

From (67), and using (70) and (71), it follows that

$$\tilde{q}_u \approx f_{\text{cor}}\left(\frac{E(e_s)}{l\,s}+1\right)\binom{m-u}{\tilde{r}-u} P_s^{\tilde{r}-u}, \quad \text{for } P_s \ll P^{(\tilde{r})}_{\dot{s},u}\,, \tag{72}$$

where

$$P^{(\tilde{r})}_{\dot{s},u} \triangleq P^{\#}_{s,u}\left(\frac{e_{s,E_s}}{l\,s}\right) = \left[f_{\text{cor}}\,\left(\frac{e_{s,E_s}}{l\,s}+1\right)\binom{m-u}{\tilde{r}-u}\right]^{-\frac{1}{\tilde{r}-u}}, \tag{73}$$

and, for $u = \tilde{r}$,

$$\tilde{q}_{\tilde{r}} = 1\,. \tag{74}$$

Remark 10: Let $P^{(\tilde{r})}_{\ddot{s},u}$ be the value of P_s for which it holds that $E(Y_{\text{UF}_{u-1}}) \approx E(Y_{\text{UF}_u})$, for $u = d+2,\ldots,\tilde{r}$. It follows from (48), (66), and (72) that

$$P^{(\tilde{r})}_{\ddot{s},u} \triangleq \frac{\lambda\, c\,(\tilde{r}-u+1)\,\tilde{n}_{u-1}}{(m-u+1)\,(u-d)\,b_{u-1}} \cdot \frac{E(X^{u-d-1})}{E(X)\,E(X^{u-d-2})} \cdot \prod_{i=1}^{u-1} V_i\,. \tag{75}$$

Corollary 2: For deterministic rebuild time distributions, it holds that

$$P^{(\tilde{r})}_{\ddot{s},\tilde{r}} < P^{(\tilde{r})}_{\ddot{s},\tilde{r}-1} < \cdots < P^{(\tilde{r})}_{\ddot{s},d+1}\,. \tag{76}$$

Proof: See Appendix C. ■

For Weibull rebuild time distributions, including exponential ones, relation (76) does not necessarily hold. Nevertheless, the following relation always holds.

Corollary 3: For Weibull rebuild time distributions, it holds that

$$P_{\ddot{s},\tilde{r}}^{(\tilde{r})} < P_{\ddot{s},u}^{(\tilde{r})} , \qquad \text{for } \; u = d+2, \ldots, \tilde{r}-1 . \tag{77}$$

Proof: See Appendix D. ■

Remark 11: Note that for $d=0$, the $P_{\ddot{s},u}^{(\tilde{r})}$ obtained from (75) is equal to $P_{\ddot{s},u}^{(\tilde{r})}$, as determined by Eq. (54) of [5]. Therefore, by considering Remarks 8 and 9 of [5] and Corollaries 2 and 3, we deduce that for the deterministic and Weibull rebuild time distributions and for any value of d, $[0, P_{\ddot{s},u}^{(\tilde{r})})$ is the range of values of P_s for which it holds that $E(Q_{\text{UF}_{\tilde{r}}}) \ll E(Q_{\text{UF}_u})$, for $u = 1, \ldots, \tilde{r}-1$, where Q is the (not effective) amount of lost user data. Consequently, EAFDL is affected when

$$P_s \gg P_{\ddot{s},\tilde{r}}^{(\tilde{r})} = P_{\ddot{s},\tilde{r}}^{(\tilde{r})} \tag{78}$$

Remark 12: According to (71) and given that $\tilde{q}_{\tilde{r}} = 1$, for $u = \tilde{r}$, approximation (72) yields $\tilde{q}_{\tilde{r}}(\text{approximation}) \approx f_{\text{cor}}\left(\frac{E(e_s)}{l\,s}+1\right) > 1 = \tilde{q}_{\tilde{r}}$. This in turn implies that $E(Y_{\text{UF}_{\tilde{r}}})/E(Y_{\text{DF}}) \approx f_{\text{cor}}\left(\frac{E(e_s)}{l\,s}+1\right) > 1$, given that $f_{\text{cor}} \geq 1$. Moreover, let $P_{\ddot{s},\tilde{r}}^{(\tilde{r})}$ be the value of P_s for which it holds that $E(Y_{\text{UF}_{\tilde{r}-1}}) \approx E(Y_{\text{DF}})$. From the above, and using (48), (53), (66), (69), (72), and (75), it follows that

$$P_{\grave{s},\tilde{r}}^{(\tilde{r})} \approx \frac{P_{\ddot{s},\tilde{r}}^{(\tilde{r})}}{f_{\text{cor}}\left(\frac{E(e_s)}{l\,s}+1\right)} < P_{\ddot{s},\tilde{r}}^{(\tilde{r})} . \tag{79}$$

Remark 13: For the deterministic and Weibull rebuild time distributions, inequalities (76), (77), and (79) imply that

$$P_{\grave{s},\tilde{r}}^{(\tilde{r})} < P_{\ddot{s},u}^{(\tilde{r})} , \qquad \text{for } \; u = d+1, \ldots, \tilde{r}-1 . \tag{80}$$

Therefore, for values of P_s in the interval $[0, P_{\grave{s},\tilde{r}}^{(\tilde{r})})$, it holds that $E(Y_{\text{UF}_u}) \ll E(Y_{\text{DF}})$, for $u = d+1, \ldots, \tilde{r}-1$. Consequently, from (64), $E(Y)$ and, by virtue of (29), EAFEL are affected when

$$P_s \gg P_{\grave{s},\tilde{r}}^{(\tilde{r})} , \tag{81}$$

where $P_{\grave{s},\tilde{r}}^{(\tilde{r})}$ is obtained using (3), (75) and (79) as follows:

$$\boxed{P_{\grave{s},\tilde{r}}^{(\tilde{r})} \triangleq \frac{\lambda\, c\;\tilde{n}_{\tilde{r}-1}\;\; E(X^{\tilde{r}-d-1})}{f_{\text{cor}}\left(\frac{E(e_s)}{l\,s}+1\right)\; l\;(\tilde{r}-d)\; b_{\tilde{r}-1}\; E(X)\, E(X^{\tilde{r}-d-2})} \prod_{i=1}^{\tilde{r}-1} V_i .} \tag{82}$$

Corollary 4: For small values of P_s such that $P_s \ll \min\left(P_{\ddot{s},u}^{(\tilde{r})}, P_{\grave{s},u}^{(\tilde{r})}\right)$, the following relation holds

$$E(Y_{\text{UF}_u}) \;\approx\; \frac{E(J)+1}{E(J)}\; P_{\text{UF}_u} . \tag{83}$$

Proof: See Appendix E. ■

3) Effective Amount of Data Loss: We proceed to derive the effective amount of lost user data during rebuild. Let $\breve{Q}$ be the amount of user data contained in the Y lost entities, which is permanently lost, too. Let also $\breve{Q}_{\text{DF}}$ and $\breve{Q}_{\text{UF}_u}$ denote the amount of lost user data associated with the direct paths $\overrightarrow{\text{DF}}$ and $\overrightarrow{UF_u}$, respectively.

Similar to (64), it holds that

$$E(\breve{Q}) \approx E(\breve{Q}_{\text{DF}}) + \sum_{u=d+1}^{\tilde{r}-1} E(\breve{Q}_{\text{UF}_u}) \approx E(\breve{Q}_{\text{DF}}) + E(\breve{Q}_{\text{UF}}) , \tag{84}$$

where $\breve{Q}_{\text{UF}}$ denotes the amount of user data lost due to unrecoverable failures with its mean given by

$$E(\breve{Q}_{\text{UF}}) \approx \sum_{u=d+1}^{\tilde{r}-1} E(\breve{Q}_{\text{UF}_u}) . \tag{85}$$

Proposition 3: For $u = d+1, \ldots, \tilde{r}-1$, it holds that

$$E(\breve{Q}_{\text{UF}_u}) \;\approx\; \frac{C}{E(J)}\;\frac{P_u}{u-d}\;\left(\prod_{j=1}^{u-1} V_j\right)\breve{q}_u , \tag{86}$$

where the expected amount $\breve{q}_u$ of lost user data of an arbitrary entity is determined by

$$\breve{q}_u \;=\; \sum_{j=1}^{E_s} e_{s,j}\;\tilde{q}_{s,u}\left(\frac{e_{s,j}}{l\,s}\right) v_j . \tag{87}$$

It also holds that

$$E(\breve{Q}_{\text{DF}}) \;\approx\; \frac{C}{E(J)}\;\frac{P_{\text{DF}}}{\tilde{r}-d}\;\left(\prod_{j=1}^{\tilde{r}-1} V_j\right)\breve{q}_{\tilde{r}} , \tag{88}$$

where C is determined by (2), $E(J)$ is determined by (16), $\tilde{q}_{s,u}(x)$ is determined by (68), P_u is determined by (48), P_{DF} is determined by (53), and V_j are determined by (37), (40), and (43).

Proof: Equations (86) and (87) are obtained in Appendix B. Equation (88) is obtained from (86) by setting $u = \tilde{r}$ and recognizing that $P_{\tilde{r}} = P_{\text{DF}}$. ■

Remark 14: For $u = d+1, \ldots, \tilde{r}-1$ and for small values of P_s, it follows from (87), and using (70), that

$$\breve{q}_u \;\approx f_{\text{cor}}\left(\frac{E({e_s}^2)}{l\,s}+E(e_s)\right)\binom{m-u}{\tilde{r}-u} P_s^{\tilde{r}-u}, \;\; P_s \ll P_{\grave{s},u}^{(\tilde{r})} , \tag{89}$$

where $P_{\grave{s},u}^{(\tilde{r})}$ is determined by (73). Moreover, for $u = \tilde{r}$ and using (13) and (71), (87) yields

$$\breve{q}_{\tilde{r}} \;=\; E(e_s) . \tag{90}$$

Also, from (72) and (89), it follows that

$$\breve{q}_u \;\approx\; f(e_s)\, E(e_s)\; \tilde{q}_u , \tag{91}$$

where

$$f(e_s) \;\triangleq\; \frac{\frac{E({e_s}^2)}{l\,s}+E(e_s)}{\left(\frac{E(e_s)}{l\,s}+1\right)E(e_s)} \;\geq\; 1 , \tag{92}$$

with the inequality being deduced from the fact that for any random variable X, it holds that $E(X^2) \geq E(X)^2$.

Combining (66), (86), and (91) yields

$$E(\breve{Q}_{\text{UF}_u}) \approx f(e_s)\, E(e_s)\, E(Y_{\text{UF}_u}) , \tag{93}$$

Also, from (69), (88), and (90), it follows that

$$E(\breve{Q}_{\text{DF}}) \approx E(Y_{\text{DF}})\, E(e_s) . \tag{94}$$

Remark 15: From Remark 10 and (93), it follows that $E(\breve{Q}_{\text{UF}_u}) \approx E(\breve{Q}_{\text{UF}_{u-1}})$ for $P_s = P_{\breve{s},u}^{(\tilde{r})}$, which is determined by (75).

Remark 16: According to (90) and given that $\breve{q}_{\tilde{r}} = E(e_s)$, for $u = \tilde{r}$, approximation (89) yields $\breve{q}_{\tilde{r}}(\text{approximation}) \approx f_{\text{cor}}\left(\frac{E({e_s}^2)}{l\,s} + E(e_s)\right) > E(e_s) = \tilde{q}_{\tilde{r}}$. This in turn implies that $E(\breve{Q}_{\text{UF}_{\tilde{r}-1}})/E(\breve{Q}_{\text{DF}}) \approx f_{\text{cor}}\left(\frac{E({e_s}^2)}{l\,s} + E(e_s)\right)/E(e_s) > 1$, given that $f_{\text{cor}} \geq 1$. Moreover, let $P_{\breve{s},\tilde{r}}^{(\tilde{r})}$ be the value of P_s for which it holds that $E(\breve{Q}_{\text{UF}_{\tilde{r}-1}}) \approx E(\breve{Q}_{\text{DF}})$. From the above, and using (48), (53), (75), (79), (86), (88), (89), and (92), it follows that

$$P_{\breve{s},\tilde{r}}^{(\tilde{r})} \approx \frac{E(e_s)\, P_{\breve{s},\tilde{r}}^{(\tilde{r})}}{f_{\text{cor}}\left(\frac{E({e_s}^2)}{l\,s} + E(e_s)\right)} \approx \frac{P_{\grave{s},\tilde{r}}^{(\tilde{r})}}{f(e_s)} \leq P_{\grave{s},\tilde{r}}^{(\tilde{r})} < P_{\breve{s},\tilde{r}}^{(\tilde{r})} . \tag{95}$$

Remark 17: For the deterministic and Weibull rebuild time distributions, inequalities (76), (77), and (98) imply that

$$P_{\breve{s},\tilde{r}}^{(\tilde{r})} < P_{\breve{s},u}^{(\tilde{r})} , \qquad \text{for } u = d+1, \ldots, \tilde{r}-1 . \tag{96}$$

Therefore, for values of P_s in the interval $[0, P_{\breve{s},\tilde{r}}^{(\tilde{r})})$, it holds that $E(\breve{Q}_{\text{UF}_u}) \ll E(\breve{Q}_{\text{DF}})$, for $u = d+1, \ldots, \tilde{r}-1$. Consequently, from (84), $E(\breve{Q})$ and, by virtue of (31), EAFEDL are affected when

$$P_s \gg P_{\breve{s},\tilde{r}}^{(\tilde{r})} , \tag{97}$$

where $P_{\breve{s},\tilde{r}}^{(\tilde{r})}$ is obtained using (3), (75) and (79) as follows:

$$\boxed{P_{\breve{s},\tilde{r}}^{(\tilde{r})} \triangleq \frac{E(e_s)\ \left(\prod_{i=1}^{\tilde{r}-1} V_i\right) \lambda\, c\ \tilde{n}_{\tilde{r}-1}\ E(X^{\tilde{r}-d-1})}{f_{\text{cor}}\left(\frac{E({e_s}^2)}{l\,s} + E(e_s)\right) l\ (\tilde{r}-d)\ b_{\tilde{r}-1}\ E(X)\, E(X^{\tilde{r}-d-2})}} . \tag{98}$$

Corollary 5: For small values of P_s such that $P_s \ll P_{\breve{s},\tilde{r}}^{(\tilde{r})}$, the fraction of lost entities $E(Y)/N_E$ reflects the fraction of lost user data $E(\breve{Q})/U$ and therefore it holds that EAFEL $\approx$ EAFEDL, which is determined by

$$\text{EAFEDL} \approx \frac{m\, P_{\text{DF}}}{n\ (\tilde{r}-d)\ E(T)} \left(\prod_{j=1}^{\tilde{r}-1} V_j\right) , \ \text{ for } \ P_s \ll P_{\breve{s},\tilde{r}}^{(\tilde{r})} . \tag{99}$$

Moreover, the common value of the EAFEL and EAFEDL reliability metrics does not depend on the entity sizes nor the symbol size.

Proof: From (64), (84), and according to Remark 16, we deduce that, for $P_s \ll P_{\breve{s},\tilde{r}}^{(\tilde{r})}$, it holds that $E(Y) \approx E(Y_{\text{DF}})$ and $E(\breve{Q}) \approx E(\breve{Q}_{\text{DF}})$. Consequently, combining (29), (30), (31), and (94), yields $E(Y)/N_E \approx E(\breve{Q})/U$ and EAFEL $\approx$ EAFEDL. Moreover, substituting (88) into (31), and using (2) and (16), yields (99). From (28), (37), (40), (43), and (53), we deduce that all variables involved in (99) are independent of the symbol size s and the entity sizes $e_{s,1}, \cdots, e_{s,E_s}$. ■

TABLE III. PARAMETER VALUES

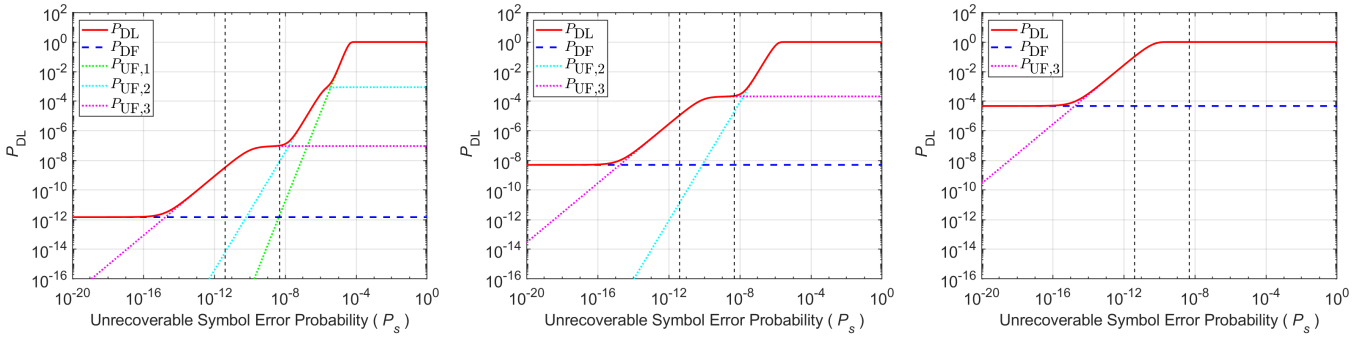
Parameter	Definition	Values
n	number of storage devices	64
c	amount of data stored on each device	20 TB
s	symbol (sector or data set) size	512 B, 5 MB
λ^{-1}	mean time to failure of a storage device	876,000 h
b	rebuild bandwidth per device	100 MB/s
m	symbols per codeword	16
l	user-data symbols per codeword	13, 14, 15
d	lazy rebuild threshold ($0 \leq d < m-l$)	0, 1, 2
U	amount of user data stored in the system	1.04 to 1.2 PB
μ^{-1}	time to read an amount c of data at a rate b from a storage device	55.5 h

VI. NUMERICAL RESULTS

Here, we assess the reliability of the clustered and declustered placement schemes for the system and the parameter values considered in [7], as listed in Table III. The system is comprised of $n = 64$ devices (HDDs), it is protected by MDS erasure codes with $m = 16$ and $l = 13$, 14, 15 and employs a lazy rebuild scheme with a threshold $d = 0$, 1, and 2. Each HDD stores an amount of $c = 20$ TB with a sector (symbol) size s of 512 bytes. The value for the parameter λ^{-1} is chosen to be $876,000$ h (100 years) that corresponds to an AFR of 1%. Also, for an average reserved rebuild bandwidth b of 100 MB/s, the mean rebuild time of a device is $E(X) = c/b = 55.5$ h, such that $\lambda/\mu = 6.3 \times 10^{-5} \ll 1$, which, according to (34), is a condition that ensures the accuracy of the reliability results obtained. Moreover, it is assumed that the maximum network rebuild bandwidth is sufficiently large ($B_{\max} \geq n\,b = 6.4$ GB/s), that the rebuild time distribution is deterministic, such that $E(X^k) = [E(X)]^k$, and that sector errors are correlated with $\bar{B} \approx 1$. From (6), it follows that $f_{\text{cor}} \approx 1$, which implies that the obtained results also apply to the case of independent sector errors.

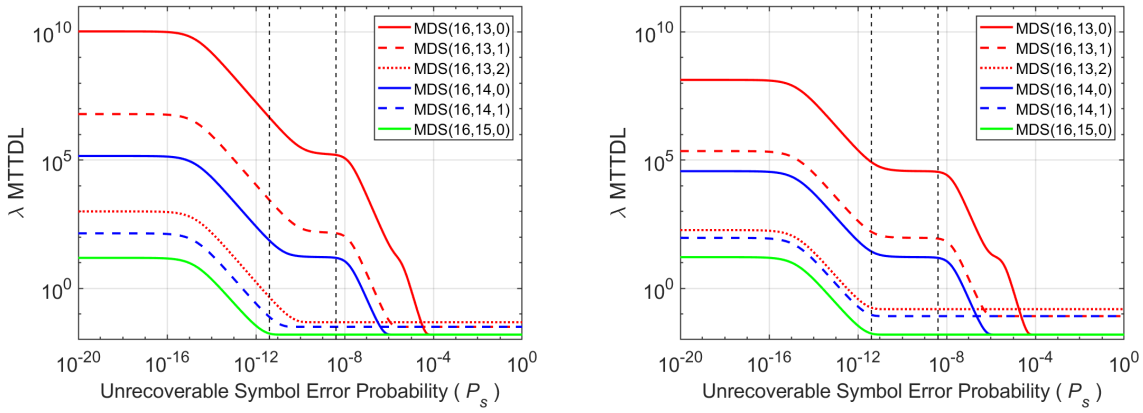
The probability of data loss P_{DL}, which does not depend on the entity size, is determined by (49) as a function of P_s and shown in Figure 8 for the declustered placement scheme ($k = n = 64$) for various MDS-coded configurations with $m = 16$, $l = 13$, and varying values of d. The probabilities P_{UF_u} and P_{DF} are also shown, as obtained from (50) and (53), respectively. We observe that P_{DL} increases monotonically with P_s and, according to Remark 5, exhibits a number of $\tilde{r} - d$ plateaus. For $d = 0$, the four plateaus are obtained from (59) and (61) as follows: $[0, 1.75\times10^{-15})$, $(1.1\times10^{-10}, 1.58\times10^{-8})$, $(1.54\times10^{-6}, 3.68\times10^{-6})$, and $(3.83\times10^{-5}, 1]$. For $d = 1$, the 3 plateaus are $[0, 1.75\times10^{-15})$, $(7.33\times10^{-11}, 1.58\times10^{-8})$, and $(1.09\times10^{-6}, 1]$. For $d = 2$, the two plateaus are $[0, 1.75\times10^{-15})$, and $(3.66\times10^{-11}, 1]$. In the interval $[4.096\times10^{-12}, 4.096\times10^{-9}]$ of practical importance for P_s, which is indicated between the two vertical dashed lines, the probability of data loss P_{DL} and, by virtue of (27), the MTTDL are degraded by one order of magnitude.

Next, we assess the reliability for the declustered placement scheme ($k = n = 64$) for the MDS-coded configurations



(a) $d = 0$ (b) $d = 1$ (c) $d = 2$

Figure 8. Probability of data loss P_{DL} vs. P_s for $d = 0, 1, 2$; $m = 16$, $l = 13$, $(\tilde{r} = 4)$, $n = k = 64$, $\lambda/\mu = 0.00006$, $c = 20$ TB, and $s = 512$ B.



(a) $k = 64$ (declustered data placement scheme) (b) $k = 16$ (clustered data placement scheme)

Figure 9. Normalized MTTDL vs. P_s for various MDS(m, l, d) codes; $n = 64$, $\lambda/\mu = 0.00006$, $c = 20$ TB, and $s = 512$ B.

considered in [7] with $m = 16$ and varying values of l and d. These configurations are denoted by MDS(m,l,d) and the corresponding results are shown in Figures 9, 10, and 11 by solid lines for $d = 0$ (no lazy rebuild employed), dashed lines for $d = 1$ and dotted lines for $d = 2$. Six configurations are considered: MDS(16,13,0), MDS(16,13,1), MDS(16,13,2), MDS(16,14,0), MDS(16,14,1), and MDS(16,15,0), for each of the declustered and clustered data placement schemes. In particular, for the clustered placement scheme, the MDS(16,15,0) and MDS(16,14,0) configurations correspond to the RAID-5 and RAID-6 systems.

The normalized λ MTTDL measure, which does not depend on the entity size, is obtained from (27) as a function of P_s and shown in Figure 9(a) for the declustered data placement scheme. The MTTDL for the MDS(16,13,0), MDS(16,13,1), and MDS(16,13,2) configurations is depicted by the red curves and is obtained from the probability of data loss shown in Figure 8. We observe that MTTDL decreases monotonically with P_s and, according to Remark 5, exhibits $\tilde{r} - d$ plateaus. In the interval of interest for P_s, MTTDL is degraded by orders of magnitude. Increasing the number of parities (reducing l) improves reliability by orders of magnitude. By contrast, employing lazy rebuild degrades reliability by orders of magnitude. Moreover, for equivalent systems, such as MDS(16,15,0), MDS(16,14,1) and MDS(16,13,2), MTTDL increases as d increases. We call *equivalent systems* those that employ a given codeword length m and have the same number $m - l - d$ of exposure levels at which the rebuild process is active.

The normalized λ MTTDL measure for the clustered data placement scheme is shown in Figure 9(b). We observe that the declustered placement scheme achieves a significantly higher MTTDL than the clustered one.

The normalized EAFEL$/\lambda$ reliability metric corresponding to the declustered data placement scheme is obtained from (29) and shown in Figure 10(a) for a fixed entity size of $e_s = 10$ GB. In the interval $[10^{-15}, 10^{-12}]$ of interest for P_{b}, EAFEL is degraded by orders of magnitude. Note that in the case of fixed-size entities, the values of the EAFEL and EAFEDL metrics are the same, because the fraction of lost entities reflects the fraction of lost user data.

Next, we consider the case of a discrete bimodal distribution for the entity size, with $e_{s,1} = 1$ MB, $e_{s,2} = 1$ TB, and probabilities $v_1 \approxeq 0.99$ and $v_2 \approxeq 0.01$ chosen such that the average entity size $E(e_s)$ is $v_1\, e_{s,1} + v_2\, e_{s,2} = 10$ GB, which is the same as the entity size e_s in the fixed-entity-size case considered previously. From (16), it follows that the average shard size $E(J)$ remains the same, which, according to (30), implies that the number N_E of entities in the system remains the same as in the fixed-entity-size case. The resulting EAFEL is shown in Figure 10(b). Comparing the case of bimodal entity sizes with that of fixed entity sizes, we observe that, for $P_{\text{b}} < 10^{-14}$, reliability remains essentially the same, whereas for higher values of P_{b}, EAFEL is reduced. The reason for that

is the following. For very small values of P_{b}, there can be at most one codeword lost, which results in one lost entity. Thus, the fraction of lost entities is $1/N_E$ in both cases. However, the lost entity in the fixed case has a size of 10 GB which is different from that of the lost entity in the bimodal case, which is either 1 MB or 1 TB. In fact, the size of the lost entity in the bimodal case is almost surely 1 TB, because the probability of this event is $v_2\, e_{s,2}/E(e_s) \approx 1$. Consequently, the size of 1 TB of the lost entity in the bimodal case is 100 times larger than that of 10 GB of the entity lost in the fixed case. This is reflected in Figure 10(c) that shows the EAFEDL metric. Note that for $P_{\mathrm{b}} = 10^{-15}$, indicated by the left vertical dashed line, EAFEDL is about 100 times larger than EAFEL. Consequently, in the case of variable size entities, it is more appropriate to consider the EAFEDL rather than the EAFEL metric, because it captures the amount of lost user data. Also, Figures 10(b) and 10(c) confirm Corollary 5 according to which, for small values of P_{b} such that $P_{\mathrm{b}} \ll P_{\check{s},\tilde{r}}^{(\tilde{r})}/s$, the EAFEL and EAFEDL metrics tend to the same value. This holds because, when $P_{\mathrm{b}} = 0$, the fraction of lost entities reflects the fraction of lost user data.

Clearly, the vulnerability of entities to loss increases with their size, which implies that lost entities are most likely large rather than small. For the case of the bimodal entity sizes, and for $v_2 \cong 0.01$, the number of the large 1-TB entities is significantly smaller than that of the 1-MB entities. We therefore deduce that the fraction of lost entities in the bimodal case is smaller than that for the fixed case, and this is more pronounced for larger values of P_{b}, as it is reflected by the EAFEL metric. By contrast, EAFEDL is larger in the bimodal case compared to the fixed case for the entire range of bit error rates. We therefore deduce that increasing the variability of the entity sizes, while keeping their average constant, results in degraded EAFEDL, but improved EAFEL, which is misleading. Clearly, the EAFEL metric that assesses the fraction of lost entities does not account for their size and the corresponding amount of lost user data and this led us to introduce the EAFEDL metric.

By observing Figures 11(a), 11(b) 11(c) that show the reliability results for the case of clustered placement, we arrive to the same conclusions. From the above discussion, it follows that in the case of variable size entities, it is important to consider the EAFEDL rather than the EAFEL metric.

The expected fraction of lost entities $E(Y)/N_E$ is obtained from (64) and shown in Figure 12 for the declustered placement scheme ($k = n = 64$) for various MDS-coded configurations with $m = 16$, $l = 13$, and varying values of d. The expected fractions of lost entities $E(Y_{\mathrm{UF}_u})/N_E$ and $E(Y_{\mathrm{DF}})/N_E$ are also shown as determined by (65) and (69), respectively. We observe that each of the $E(Y_{\mathrm{UF}_u})/N_E$ curves exhibits two plateaus owing to the bimodal nature of the entity sizes. According to Remark 13, $E(Y)$ and EAFEL degrade when P_s is greater than $P_{\check{s},\tilde{r}}^{(\tilde{r})}$, which for deterministic rebuild times and in the absence of network rebuild bandwidth constraints, by virtue of (82), is equal to 1.3×10^{-13}. For a symbol size of 512 bytes, the corresponding unrecoverable bit error probability is $P_{\mathrm{b}} \approx P_{\check{s},\tilde{r}}^{(\tilde{r})} \,/\, (512 \times 8) = 1.3 \times 10^{-13} \,/\, 4096 = 3.18 \times 10^{-17}$. This is depicted by the red curves in Figures 12 and 10(b).

The expected fraction of the effective amount of lost user data $E(\breve{Q})/U$ is obtained from (84) and shown in Figure 13. The expected fractions of the effective amounts of lost user data $E(\breve{Q}_{\mathrm{UF}})/U$ and $E(\breve{Q}_{\mathrm{DF}})/U$ are also shown as determined by (85) and (88), respectively. Despite the bimodal nature of the entity sizes, we observe that in this case each of the $E(\breve{Q}_{\mathrm{UF}})/U$ curves exhibits only a single plateau. According to Remark 17, $E(\breve{Q})$ and EAFEDL degrade when P_s is greater than $P_{\check{s},\tilde{r}}^{(\tilde{r})}$, which for deterministic rebuild times and in the absence of network rebuild bandwidth constraints, by virtue of (98), is equal to 1.3×10^{-15}. For a symbol size of 512 bytes, this degradation occurs when the unrecoverable bit error probability P_{b} is greater than $P_{\check{s},\tilde{r}}^{(\tilde{r})} \,/\, (512 \times 8) = 1.3 \times 10^{-15} \,/\, 4096 = 3.18 \times 10^{-19}$. This is depicted by the red curves in Figures 13 and 10(c). Note also that for extremely small values of P_{b}, such that $P_{\mathrm{b}} \ll 3.18\times10^{-19}$, and according to Corollary 5, it holds that $E(\breve{Q})/U \approx E(Y)/N_E$. This also holds when $P_{\mathrm{b}} \to 1$.

The effect of symbol size on reliability is assessed by considering the case of a large 5-MB symbol size. The corresponding normalized EAFEL$/\lambda$ and EAFEDL$/\lambda$ reliability metrics are shown in Figures 14 and 15. As expected, comparing these results with those shown in Figures 10 and 11, system reliability degrades compared to the case of a smaller symbol size. This degradation applies to both the EAFEL and EAFEDL reliability metrics.

Next, we assess the system reliability for the CERN file size distribution [22] that was considered in [23] and listed in Table IV shown in Appendix B. For the file sizes uniformly distributed within the bins, the mean is 843 MB, the standard deviation is 2.8 GB, the second moment is 8.9 GB^2 and the coefficient of variation is equal to 3.39. It turns out that the reliability metrics are extremely well approximated by considering the file sizes $e_{s,j}$ to be the bin mean sizes, such that $E_s = 38$. In this case, the mean is 843 MB, the standard deviation is 2.8 GB, the second moment is 8.5 GB^2 and the coefficient of variation is 3.37. The corresponding reliability results are shown in Figures 16 and 17. In all cases considered, the reliability level achieved by the declustered data placement scheme is higher than that of the clustered one.

VII. Real-World Erasure Coding Schemes

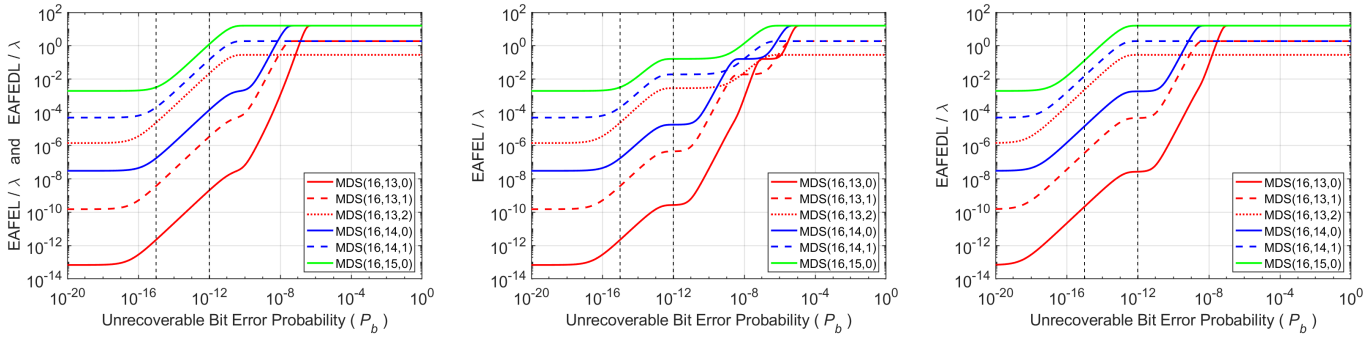
Here we assess the reliability of systems that store files whose size is distributed according to the CERN distribution listed in Table IV and shown in Figure 24(a). In particular, we assess the reliability of the practical systems considered in [4] that store an amount of $U = 1.2$ PB user data on devices (disks) whose capacity is $c = 20$ TB. This amount of user data can therefore be stored on $U/c = 60$ devices. The system comprises n devices, where n is determined using (1) as follows:

$$n = \frac{U}{c}\,\frac{m}{l} = 60\,\frac{m}{l}\,. \tag{100}$$

Subsequently, we consider the following real-world erasure coding schemes:

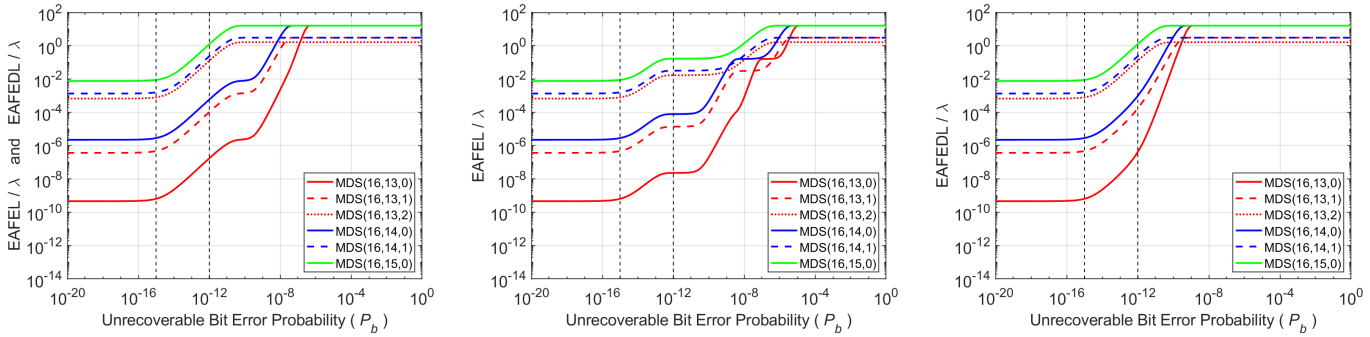
1) the 3-way replication (triplication) scheme that was initially used by Google's GFS, Microsoft® Azure[1], and

[1]Microsoft is a trademark of Microsoft Corporation in the United States, other countries, or both.



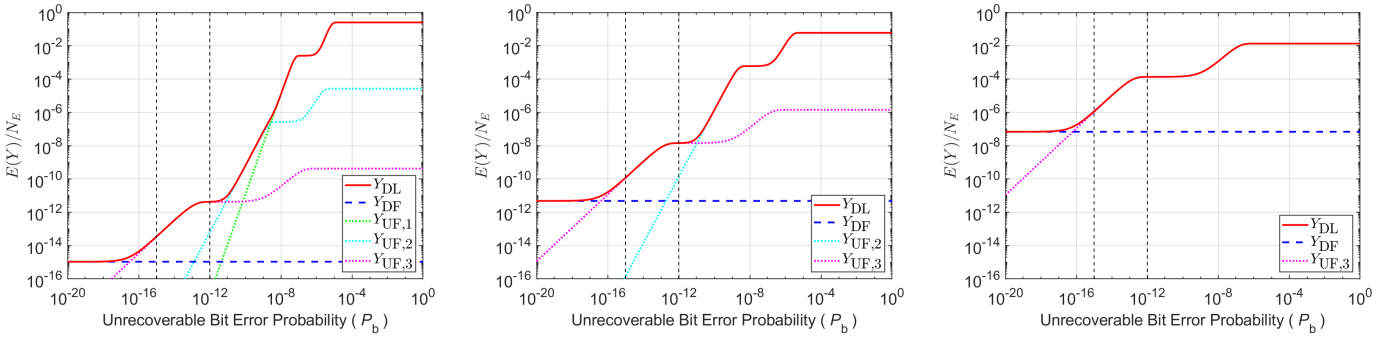
(a) Fixed Entity Size: $e_s = 10$ GB (b) Bimodal Entity Sizes: $e_{s,1} = 1$ MB, $e_{s,2} = 1$ TB (c) Bimodal Entity Sizes: $e_{s,1} = 1$ MB, $e_{s,2} = 1$ TB

Figure 10. Normalized EAFEL and EAFEDL vs. P_{b} for various MDS(m, l, d) codes; symbol size $s = 512$ B, declustered data placement.



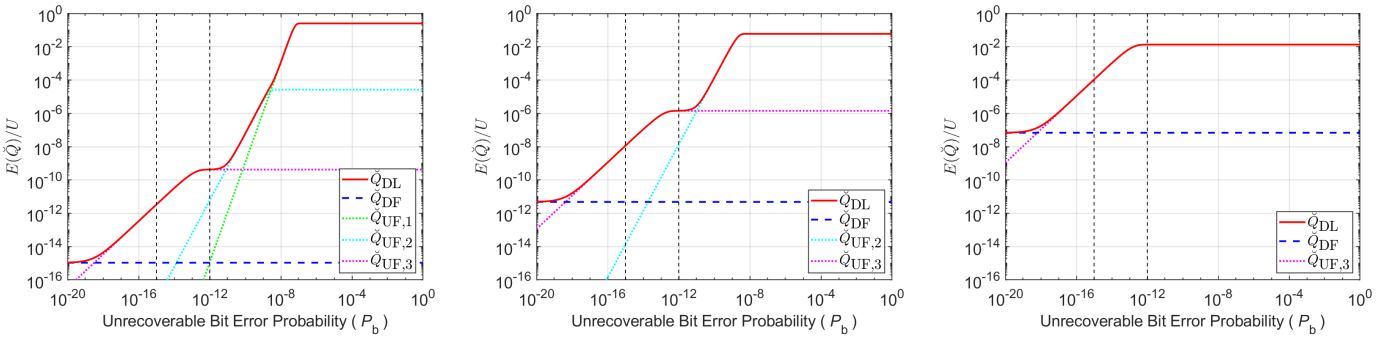
(a) Fixed Entity Size: $e_s = 10$ GB (b) Bimodal Entity Sizes: $e_{s,1} = 1$ MB, $e_{s,2} = 1$ TB (c) Bimodal Entity Sizes: $e_{s,1} = 1$ MB, $e_{s,2} = 1$ TB

Figure 11. Normalized EAFEL and EAFEDL vs. P_{b} for various MDS(m, l, d) codes; symbol size $s = 512$ B, clustered data placement.



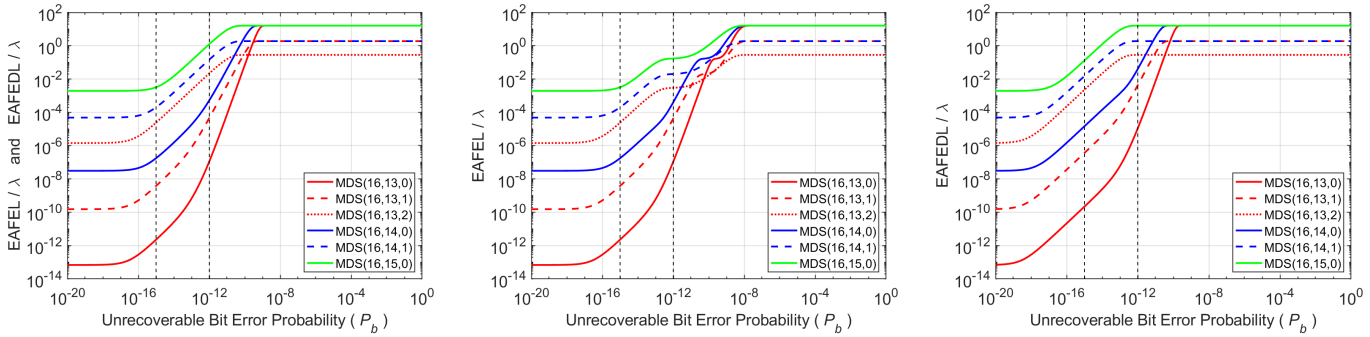
(a) $d = 0$ (b) $d = 1$ (c) $d = 2$

Figure 12. Normalized $E(Y)$ vs. P_s for $d = 0, 1, 2$; $m = 16$, $l = 13$, $(\tilde{r} = 4)$, $n = k = 64$, $c = 20$ TB, and $s = 512$ B, bimodal entity sizes.



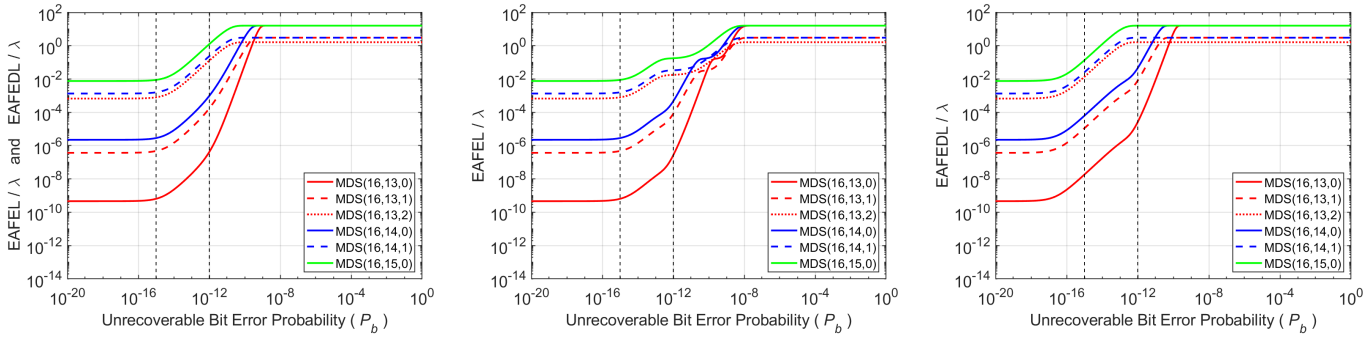
(a) $d = 0$ (b) $d = 1$ (c) $d = 2$

Figure 13. Normalized $E(\breve{Q})$ vs. P_s for $d = 0, 1, 2$; $m = 16$, $l = 13$, $(\tilde{r} = 4)$, $n = k = 64$, $c = 20$ TB, and $s = 512$ B, bimodal entity sizes.



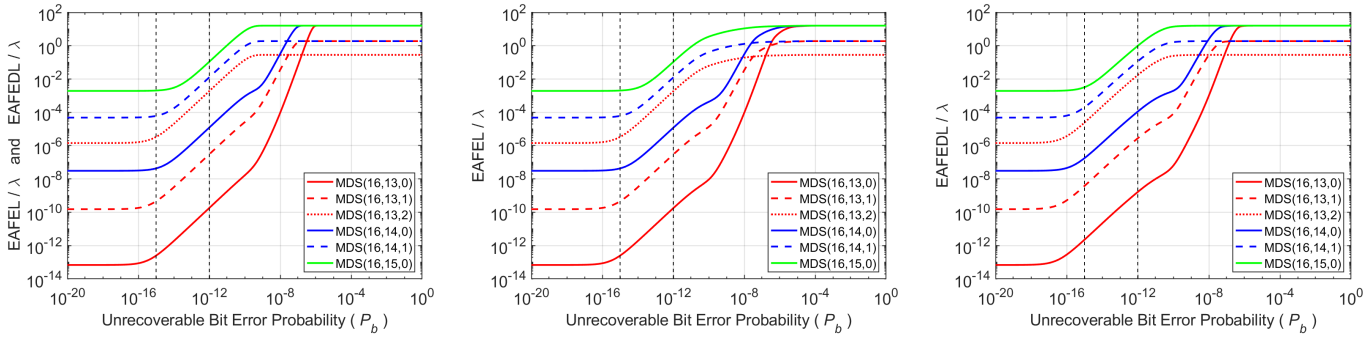
(a) Fixed Entity Size: $e_s = 10$ GB (b) Bimodal Entity Sizes: $e_{s,1} = 1$ MB, $e_{s,2} = 1$ TB (c) Bimodal Entity Sizes: $e_{s,1} = 1$ MB, $e_{s,2} = 1$ TB

Figure 14. Normalized EAFEL and EAFEDL vs. P_{b} for various MDS(m, l, d) codes; symbol size $s = 5$ MB, declustered data placement.



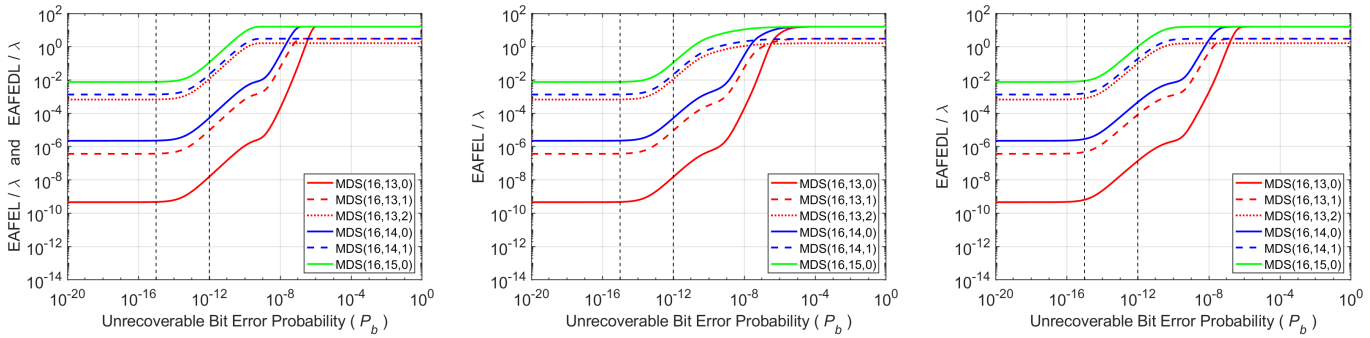
(a) Fixed Entity Size: $e_s = 10$ GB (b) Bimodal Entity Sizes: $e_{s,1} = 1$ MB, $e_{s,2} = 1$ TB (c) Bimodal Entity Sizes: $e_{s,1} = 1$ MB, $e_{s,2} = 1$ TB

Figure 15. Normalized EAFEL and EAFEDL vs. P_{b} for various MDS(m, l, d) codes; symbol size $s = 5$ MB, clustered data placement.



(a) Fixed File Size: $e_s = 843$ MB (b) CERN File Sizes; $E(e_s) = 843$ MB (c) CERN File Sizes; $E(e_s) = 843$ MB

Figure 16. Normalized EAFEL and EAFEDL vs. P_{b} for various MDS(m, l, d) codes; symbol size $s = 512$ B, declustered data placement.



(a) Fixed File Size: $e_s = 843$ MB (b) CERN File Sizes; $E(e_s) = 843$ MB (c) CERN File Sizes; $E(e_s) = 843$ MB

Figure 17. Normalized EAFEL and EAFEDL vs. P_{b} for various MDS(m, l, d) codes; symbol size $s = 512$ B, clustered data placement.

Facebook. In this case, $m = 3$, $l = 1$, with a corresponding storage efficiency of $s_{\text{eff}} = 33\%$. According to (100), this scheme requires the employment of $n = 180$ devices.

2) the RS(9,6) erasure coding scheme employed by Google's GFS as well as QFS [24][25], which for $m = 9$ and $l = 6$ achieves a storage efficiency of $s_{\text{eff}} = 66\%$ and requires a number of $n = 90$ devices.
3) the MDS(16,12) erasure coding scheme akin to the LRC(16,12) code used by Microsoft® Azure [26], which for $m = 16$ and $l = 12$ achieves a storage efficiency of $s_{\text{eff}} = 75\%$ and requires a number of $n = 80$ devices.
4) the RS(14,10) erasure coding scheme employed by Facebook [27], which for $m = 14$ and $l = 10$ achieves a storage efficiency of $s_{\text{eff}} = 71\%$ and requires a number of $n = 84$ devices.

We proceed to assess the reliability of the four erasure coding schemes assuming a 512-B symbol size and for the declustered data placement scheme, which achieves a superior data reliability. The results for the EAFEL and EAFEDL reliability metrics are shown in Figures 18 and 19, respectively. We observe that, in all cases, EAFEDL is larger than EAFEL.

First, we assess the reliability of the 3-way replication (triplication) scheme. Figure 19(a) shows that, in the interval of interest for P_{b}, EAFEDL ranges between 10^{-12} and 10^{-9}. In particular, when P_{b} is larger than 10^{-14}, EAFEDL is larger than 10^{-11}, the durability of eleven nines (11 9s) targeted by the Amazon S3 [28]. Employing the MDS(9,6) coding scheme, improves reliability by orders of magnitude. Figure 19(b) shows that, in the interval of interest for P_{b}, EAFEDL ranges between 10^{-16} and 10^{-13}. Further reliability improvement is achieved by employing the MDS(16,12) coding scheme. According to Figure 19(c), in the interval of interest for P_{b}, EAFEDL ranges between 10^{-20} and 10^{-17}. Superior reliability is achieved by employing the MDS(14,10) coding scheme. Figure 19(d) shows that, in the interval of interest for P_{b}, EAFEDL ranges between 10^{-21} and 10^{-18}.

Also, Figures 18 and 19 confirm Corollary 5 according to which, for small values of P_{b} such that $P_{\text{b}} \ll P_{\check{s},\check{r}}^{(\check{r})}/s$, the EAFEL and EAFEDL metrics tend to the same value. However, in the interval of interest for P_{b}, by employing the 3-way replication, MDS(9,6), and MDS(16,12) coding schemes, both EAFEL and EAFEDL improve by four orders of magnitude, successively. By contrast, employing the MDS(14,10) coding scheme results in a reliability improvement of only one order of magnitude of that achieved by the MDS(16,12) coding scheme.

We proceed to assess the reliability of the four erasure coding schemes for the declustered data placement scheme by considering the case of a large 5-MB symbol size. The results for the EAFEL and EAFEDL reliability metrics are shown in Figures 20 and 21, respectively. We observe that, in all cases, EAFEDL is larger than EAFEL.

First, we assess the reliability of the 3-way replication scheme. Figure 21(a) shows that, in the interval of interest for P_{b}, EAFEDL ranges between 10^{-12} and 10^{-7}. In particular, when P_{b} is larger than 10^{-14}, EAFEDL is larger than 10^{-11}, the durability of eleven nines (11 9s) targeted by the Amazon S3. Comparing Figures 18(a) and 20(a) as well as Figures 19(a) and 21(a), we observe that the increased symbol size affects EAFEL and EAFEDL only for P_{b} values in the interval $(10^{-14}, 10^{-7})$.

Employing the MDS(9,6) coding scheme, improves reliability by orders of magnitude. Figure 21(b) shows that, in the interval of interest for P_{b}, EAFEDL ranges between 10^{-16} and 10^{-10}. In particular, when P_{b} is larger than 8×10^{-13}, EAFEDL is larger than 10^{-11}, the durability of eleven nines (11 9s) targeted by the Amazon S3 [28]. Comparing Figures 18(b) and 20(b) as well as Figures 19(b) and 21(b), we observe that the increased symbol size affects EAFEL and EAFEDL only for P_{b} values in the intervals $(10^{-15}, 10^{-5})$ and $(10^{-15}, 10^{-6})$, respectively.

Further reliability improvement is achieved by employing the MDS(16,12) coding scheme. According to Figure 19(c), in the interval of interest for P_{b}, EAFEDL ranges between 10^{-20} and 10^{-13}. Comparing Figures 18(c) and 20(c) as well as Figures 19(c) and 21(c), we observe that the increased symbol size affects EAFEL and EAFEDL only for P_{b} values in the intervals $(10^{-15}, 10^{-5})$ and $(10^{-15}, 10^{-6})$, respectively.

Superior reliability is achieved by employing the MDS(14,10) coding scheme. Figure 21(d) shows that, in the interval of interest for P_{b}, EAFEDL ranges between 10^{-21} and 10^{-13}. Comparing Figures 18(d) and 20(d) as well as Figures 19(d) and 21(d), we observe that the increased symbol size affects EAFEL and EAFEDL only for P_{b} values in the interval $(10^{-15}, 10^{-5})$.

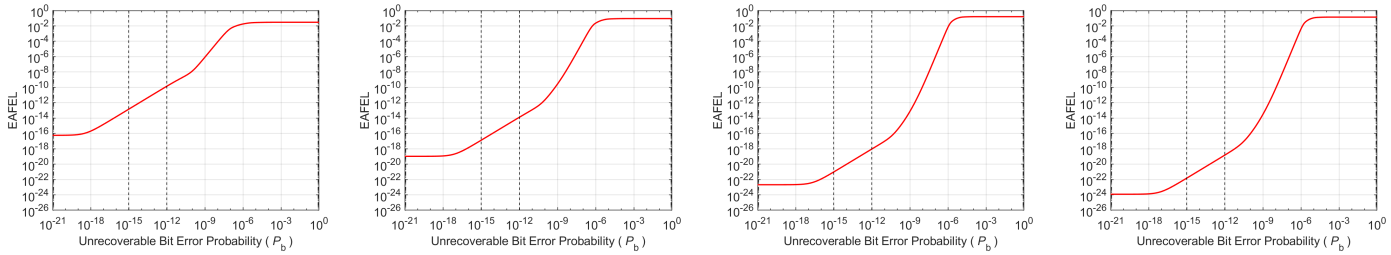
Figures 20 and 21 confirm Corollary 5 according to which, for small values of P_{b} such that $P_{\text{b}} \ll P_{\check{s},\check{r}}^{(\check{r})}/s$, the EAFEL and EAFEDL metrics tend to the same value. However, for $P_{\text{b}} = 10^{-15}$, by employing the 3-way replication, MDS(9,6), and MDS(16,12) coding schemes, both EAFEL and EAFEDL improve by four orders of magnitude, successively. By contrast, employing the MDS(14,10) coding scheme results in a reliability improvement of only one order of magnitude of that achieved by the MDS(16,12) coding scheme. Also, for $P_{\text{b}} = 10^{-12}$, by employing the 3-way replication, MDS(9,6), and MDS(16,12) coding schemes, both EAFEL and EAFEDL improve by three orders of magnitude, successively. By contrast, employing the MDS(14,10) coding scheme does not achieve any reliability improvement compared to the MDS(16,12) coding scheme.

A. Reliability Improvement

The improvement of the EAFEL and EAFEDL reliability metrics achieved by the erasure coding schemes considered over the initial 3-way replication is shown in Figures 22 and 23.

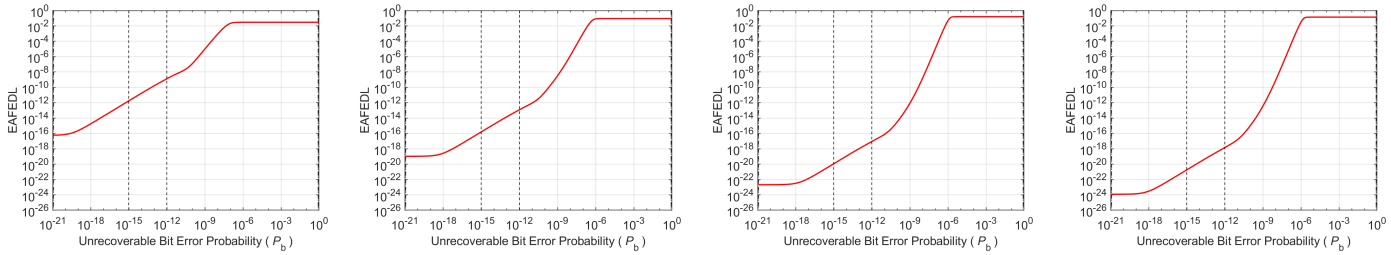
For the declustered data placement and for a symbol size of 512 B, Figure 22 demonstrates that in the interval of interest, the MDS(9,6) erasure coding scheme improves reliability by four orders of magnitude, the MDS(16,12) erasure coding scheme improves reliability by eight orders of magnitude, whereas the MDS(14,10) erasure coding scheme improves reliability by nine orders of magnitude.

For the declustered data placement and for a symbol size of 5 MB, Figure 23 demonstrates that in the interval of interest, the reliability improvement achieved by the erasure coding schemes considered varies. In particular, for $P_{\text{b}} = 10^{-15}$, the



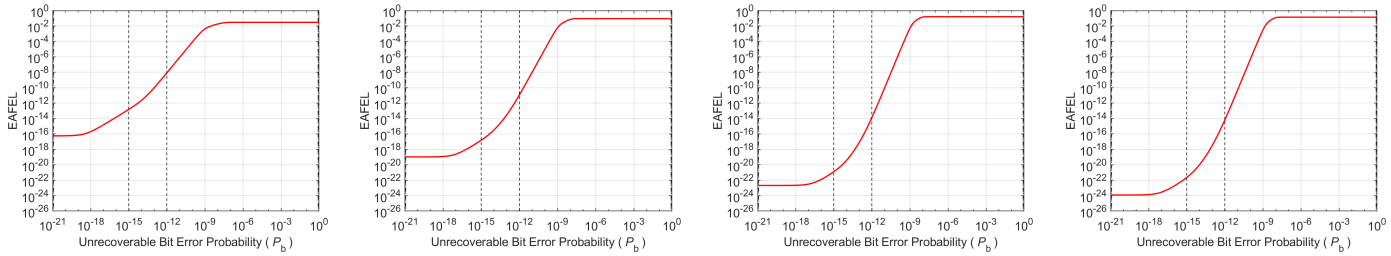
(a) MDS(3,1) (b) MDS(9,6) (c) MDS(16,12) (d) MDS(14,10)

Figure 18. EAFEL vs. P_s for various MDS coding schemes; symbol size $s = 512$ B, declustered data placement.



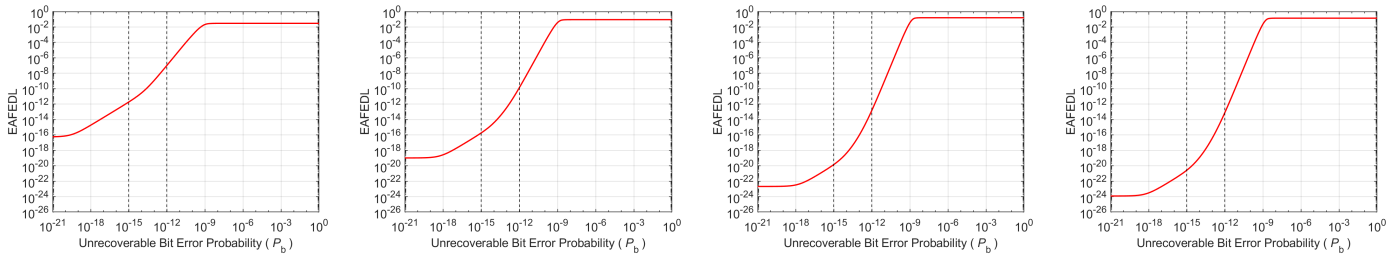
(a) MDS(3,1) (b) MDS(9,6) (c) MDS(16,12) (d) MDS(14,10)

Figure 19. EAFEDL vs. P_s for various MDS coding schemes; symbol size $s = 512$ B, declustered data placement.



(a) MDS(3,1) (b) MDS(9,6) (c) MDS(16,12) (d) MDS(14,10)

Figure 20. EAFEL vs. P_s for various MDS coding schemes; symbol size $s = 5$ MB, declustered data placement.



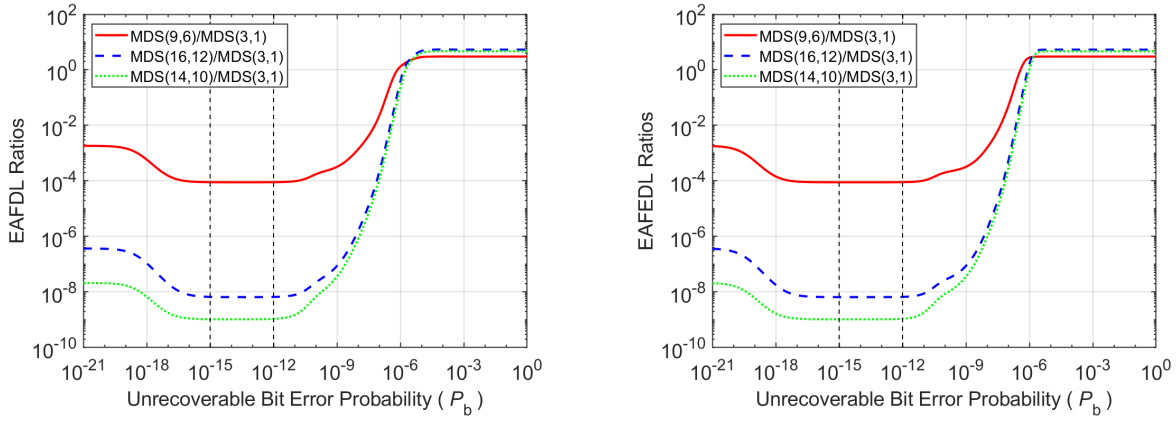
(a) MDS(3,1) (b) MDS(9,6) (c) MDS(16,12) (d) MDS(14,10)

Figure 21. EAFEDL vs. P_s for various MDS coding schemes; symbol size $s = 5$ MB, declustered data placement.

MDS(9,6) erasure coding scheme improves reliability by four orders of magnitude, the MDS(16,12) erasure coding scheme improves reliability by eight orders of magnitude, whereas the MDS(14,10) erasure coding scheme improves reliability by nine orders of magnitude. However, for $P_b = 10^{-12}$, the MDS(9,6) erasure coding scheme improves reliability by three orders of magnitude, whereas the MDS(16,12) and MDS(14,10) erasure coding scheme improve reliability by six orders of magnitude.

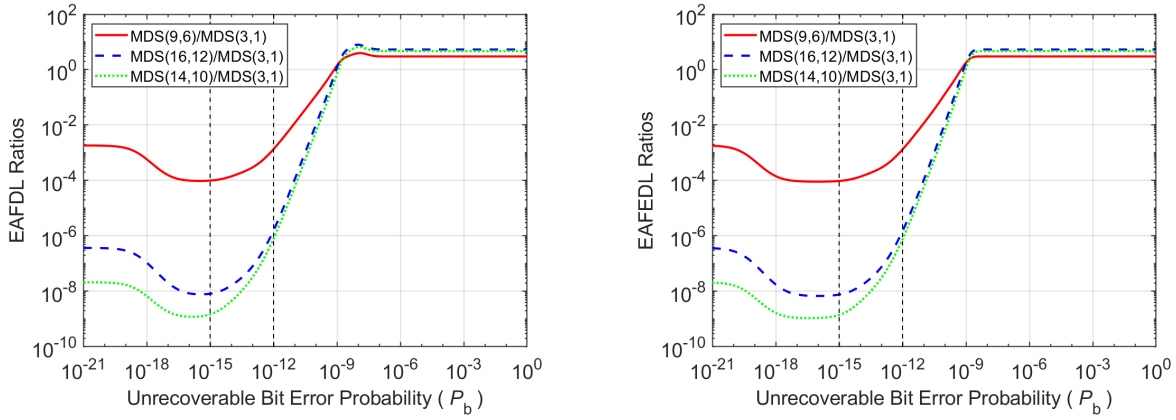
VIII. Conclusions

The Expected Annual Fraction of Entity Loss EAFEL metric assesses the durability of data storage systems at an entity, say file, object, or block level. Contrary to the Mean Time to Data Loss (MTTDL) metric, EAFEL is affected by the distribution of the number of codewords that entities span. The distribution of this number was obtained analytically in closed form for the segregated and the random entity placement cases as a function of the size of the entities and the frequency of their occurrence. It was also demonstrated that, in certain cases



(a) EAFEL ratios (b) EAFEDL ratios

Figure 22. Ratios of the EAFEL and EAFEDL metrics for the MDS(9,6), MDS(16,12), and MDS(14,10) schemes to those corresponding to the 3-way replication scheme; symbol size $s = 512$ B, declustered data placement.



(a) EAFEL ratios (b) EAFEDL ratios

Figure 23. Ratios of the EAFEL and EAFEDL metrics for the MDS(9,6), MDS(16,12), and MDS(14,10) schemes to those corresponding to the 3-way replication scheme; symbol size $s = 5$ MB, declustered data placement.

of deterministic entity placements of variable-size entities, this distribution also depends on their actual placement.

To evaluate the durability of storage systems in the case of variable-size entities, a new reliability metric was introduced, the Expected Annual Fraction of Effective Data Loss (EAFEDL), which assesses the fraction of lost user data annually at the entity level. The MTTDL, EAFEL, and EAFEDL metrics were obtained analytically for erasure-coding redundancy schemes and for the clustered, declustered, and symmetric data placement schemes. Closed-form expressions capturing the effect of unrecoverable latent errors and lazy rebuilds were derived. We established that the reliability of storage systems is adversely affected by the presence of latent errors and that the declustered data placement scheme offers superior reliability. We demonstrated that an increased variability of entity sizes results in improved EAFEL, but degraded EAFEDL. We also established that EAFEL and EAFEDL are adversely affected by the symbol size. We considered several real-world erasure coding schemes and demonstrated their efficiency. The analytical reliability results obtained enable the identification of erasure-coded redundancy schemes that ensure a desired level of reliability.

This work has the potential to be applied for further studies of data storage reliability and it is particularly relevant for tape storage reliability, which is a subject of further investigation [29].

Appendix A

Proof of Corollary 1.

From Eqs. (57) and (65) of [21], (61) of [21] yields

$$P_{\mathrm{UF}_u}(R_{d+1}) \approx -(\lambda b_{d+1} R_{d+1})^{u-d-1} \left(\prod_{i=d+1}^{u-1} \frac{\tilde{n}_i}{b_i} V_i^{u-1-i} \right) \cdot \cdot \left(\sum_{j=1}^{\infty} \frac{\log(\hat{q}_u)^j}{(u-d-1+j)!} \right). \quad (101)$$

From (55) and for $P_s \ll P^*_{s,u}$, it follows that $\hat{q}_u \approx 1$. Furthermore, $\log(\hat{q}_u) = -(1-\hat{q}_u) + O((1-\hat{q}_u)^2) \approx -(1-\hat{q}_u)$. Consequently, by virtue of (55), it holds that $\log(\hat{q}_u) \approx -Z_u P_s^{\tilde{r}-u}$. For small values of P_s, all the terms of the summation in (101) are negligible compared with the first one.

Therefore, from the above, it follows that

$$P_{\text{UF}_u}(R_{d+1}) \;\approx\; (\lambda b_{d+1} R_{d+1})^{u-d-1} \left(\prod_{i=d+1}^{u-1} \frac{\tilde{n}_i}{b_i}\, V_i^{u-1-i} \right) \cdot \cdot \frac{Z_u\, P_s^{\tilde{r}-u}}{(u-d)!} \,. \tag{102}$$

Unconditioning (102) on R_{d+1}, and using (33) and (44), yields

$$P_{\text{UF}_u} \;\approx\; \frac{(\lambda\, c \prod_{j=1}^{d} V_j)^{u-d-1}}{(u-d)!} \left(\prod_{i=d+1}^{u-1} \frac{\tilde{n}_i}{b_i}\, V_i^{u-1-i} \right) \cdot \cdot \frac{E(X^{\tilde{r}-d-1})}{[E(X)]^{\tilde{r}-d-1}}\, Z_u\, P_s^{\tilde{r}-u} \,. \tag{103}$$

Consider the direct path $\overrightarrow{UF_u} = 1 \to 2 \to \cdots \to u \to \text{UF}$. Then the probability $P_{\text{UF}_u}(R_{d+1}, \vec{\alpha}_{u-1})$ of entering exposure level u through vector $\vec{\alpha}_{u-1} \triangleq (\alpha_1, \ldots, \alpha_{u-1})$ and encountering an unrecoverable failure during the rebuild process at this exposure level, given a rebuild time R_{d+1}, is determined by [21, Eq. (46)]

$$P_{\text{UF}_u}(R_{d+1}, \vec{\alpha}_{u-1}) = P_u(R_{d+1}, \vec{\alpha}_{u-2}) \cdot P_{u\to\text{UF}}(R_{d+1}, \vec{\alpha}_{u-1}) \,. \tag{104}$$

where P_u is the probability of entering exposure level u and $P_{u\to\text{UF}}$ is the probability of encountering an unrecoverable failure during the rebuild process at this exposure level. We now proceed to calculate $P_{u\to\text{UF}}(R_{d+1}, \vec{\alpha}_{u-1})$. Upon entering exposure level u, the rebuild process attempts to restore the C_u most-exposed codewords, each of which has $m-u$ remaining symbols. The probability q_u that a codeword can be restored is determined by (52). Note that, if a codeword is corrupted, then at least one of its l user-data symbols is lost. When symbol errors are independent, codewords are independently corrupted. Consequently, the conditional probability $P_{\text{UF}|C_u}$ of encountering an unrecoverable failure during the rebuild process of the C_u codewords is determined by $1 - q_u^{C_u}$ [21, Eq. (58)]. In the case of correlated symbol errors, $P_{\text{UF}|C_u}$ is determined by $1 - q_u^{f_{\text{cor}}\, C_u}$ [5, Eq. (98)]. Consequently, it holds that

$$P_{\text{UF}|C_u} = 1 - q_u^{f_{\text{cor}}\, C_u} \,, \quad \text{for } u = d+1, \ldots, \tilde{r} \,. \tag{105}$$

Substituting (46) into (105) and using (51) yields

$$P_{u\to\text{UF}}(R_{d+1}, \vec{\alpha}_{u-1}) \approx 1 - q_u^{C \prod_{j=1}^{u-1} V_j\, \alpha_j} = 1 - \hat{q}_u^{\prod_{j=1}^{u-1} \alpha_j} \,. \tag{106}$$

Substituting (106) into (104) yields

$$P_{\text{UF}_u}(R_{d+1}, \vec{\alpha}_{u-1}) \;\approx\; P_u(R_{d+1}, \vec{\alpha}_{u-2}) \left[1 - \hat{q}_u^{\prod_{j=1}^{u-1} \alpha_j} \right] . \tag{107}$$

From (55) and for $P_s \gg P^*_{s,u}$, it follows that $\hat{q}_u \approx 0$, which by virtue of (106) implies that $P_{u\to\text{UF}}(R_1, \vec{\alpha}_{u-1}) \approx 1$. Consequently, it follows from (104) that $P_{\text{UF}_u} \approx P_u$. Also, substituting (56) into (103) yields (57), with the variable A_u determined by (58). In particular, $P_{\tilde{s},u}^{(\tilde{r})}$ is obtained from the approximation (57) $P_{\text{UF}_u} \approx A_u\, (P_{\tilde{s},u}^{(\tilde{r})})^{\tilde{r}-u} = P_u$ and using (2), (48), and (58).

□

APPENDIX B

Proof of Proposition 2.

Upon entering exposure level u ($u \geq d+1$), there are C_u most-exposed codewords to be recovered. As a shard size of s_s corresponds to J symbols, an entity size e_s corresponds to J codewords. Therefore, the average entity of size $E(e_s)$ determined by (13) corresponds to $E(J)$ codewords, with $E(J)$ determined by (16). Consequently, for the number E_u of entities to be recovered it holds that

$$E_u \;\approx\; \frac{C_u}{E(J)} \,, \quad \text{for } u = d+1, \ldots, \tilde{r}-1 \,. \tag{108}$$

Let K ($K \geq 1$) denote the number of codewords that an entity of size e_s spans or, equivalently, the number of symbols that a shard of size s_s spans. The entity is lost if any of these K codewords is permanently lost. Therefore, according to Eq. (98) of [5], the probability of recovering the entity is $q_u^{f_{\text{cor}}\, K}$, where q_u is the probability of restoring a codeword and is determined by (52), and f_{cor} accounts for the correlation of latent errors and is determined by Eq. (29) of [5]. Consequently, the probability $\tilde{q}_u|K$ of loss of an entity that spans K codewords is determined by

$$\tilde{q}_u|K \;=\; 1 - q_u^{f_{\text{cor}}\, K} \,. \tag{109}$$

Unconditioning (109) on K using (23) yields the probability $\tilde{q}_{s,u}(J)$ that the entity (for the shard size J) is lost, where $\tilde{q}_{s,u}(x)$ is determined by (68). Thus, using (4), the probability $\tilde{q}_u(e_s)$ that the entity is lost is determined by

$$\tilde{q}_u(e_s) \;=\; \tilde{q}_{s,u} \left(\frac{e_s}{l\, s} \right) \,. \tag{110}$$

For this entity, the expected amount $\breve{q}_u(e_s)$ of lost user data is

$$\breve{q}_u(e_s) \;=\; e_s\, \tilde{q}_u(e_s) \,. \tag{111}$$

From (12), the probability $\tilde{q}_u$ that an arbitrary entity is lost is

$$\tilde{q}_u = \sum_{j=1}^{E_s} \tilde{q}_u(e_{s,j})\, v_j \,, \tag{112}$$

which, using (110), yields (67).

Similarly, from (12), it follows that the expected amount $\breve{q}_u$ of lost user data of an arbitrary entity is determined by

$$\breve{q}_u = \sum_{j=1}^{E_s} \breve{q}_u(e_{s,j})\, v_j \,, \tag{113}$$

which, using (110) and (111), yields (87).

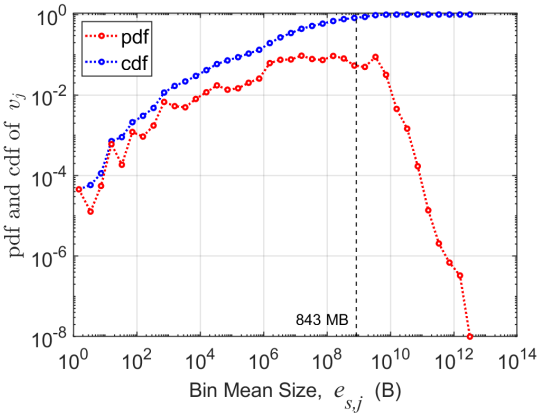
Remark 18: Note that (108) holds when $C_u \gg E(J)$. In the case where $C_u \ll E(J)$, it holds that $E_u = 1$, that is, a single entity is to be recovered. Let $\hat{e}_s$ denote its size. From the pdf of the lifetime of sampled intervals [30], we deduce that the pdf $\{\hat{v}_j\}$ of the size $\hat{e}_s$ of the sampled entity is no longer the typical $\{v_j\}$ pdf, but is determined by

$$\hat{v}_j \;=\; P(\hat{e}_s = e_{s,j}) = \frac{e_{s,j}\, v_j}{E(e_s)} \,, \quad \text{for } j = 1, 2, \ldots, E_s \,. \tag{114}$$

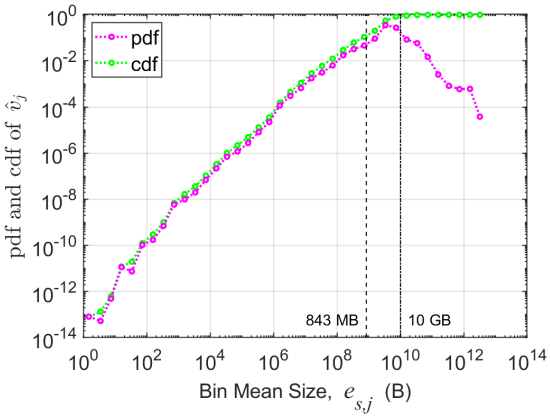
The $\{\hat{v}_j\}$ pdf of $\hat{e}_s$ is listed in Table IV and shown in Figure 24(b). For the file sizes uniformly distributed within the bins,

TABLE IV. CERN FILE SIZE e_s AND SAMPLED FILE SIZE $\hat{e}_s$ DISTRIBUTIONS

j	Bins			Bin Mean Size $e_{s,j}$	pdf v_j	pdf $\hat{v}_j$
1	1 B	–	2 B	2 B	0.00004559	0.000000000000081
2	2 B	–	5 B	4 B	0.00001275	0.000000000000053
3	5 B	–	10 B	8 B	0.00005533	0.000000000000492
4	10 B	–	22 B	16.0 B	0.00060401	0.000000000011464
5	22 B	–	46 B	34.0 B	0.00018569	0.000000000007489
6	46 B	–	100 B	73.0 B	0.00121244	0.000000000104989
7	100 B	–	215 B	157.5 B	0.00093013	0.000000000173774
8	215 B	–	464 B	339.5 B	0.00174431	0.000000000702464
9	464 B	–	1 KB	732.0 B	0.00675513	0.000000005865509
10	1 KB	–	2.154 KB	1.577 KB	0.00530524	0.000000009924249
11	2.154 KB	–	4.642 KB	3.398 KB	0.00496005	0.000000019992649
12	4.642 KB	–	10 KB	7.321 KB	0.00800625	0.000000069528117
13	10 KB	–	21.544 KB	15.772 KB	0.01174913	0.000000219813008
14	21.544 KB	–	46.416 KB	33.980 KB	0.01738480	0.000000700735281
15	46.416 KB	–	100 KB	73.208 KB	0.01359001	0.000001180155486
16	100 KB	–	215.443 KB	157.721 KB	0.01471745	0.000002753495549
17	215.443 KB	–	464.159 KB	339.801 KB	0.02018806	0.000008137296681
18	464.159 KB	–	1 MB	732.079 KB	0.02566358	0.000022286219101
19	1 MB	–	2.154 MB	1.577 MB	0.06221012	0.000116389428894
20	2.154 MB	–	4.642 MB	3.398 MB	0.07519022	0.000303072948937
21	4.642 MB	–	10 MB	7.321 MB	0.07654035	0.000664675346806
22	10 MB	–	21.544 MB	15.772 MB	0.09501620	0.001777665788444
23	21.544 MB	–	46.416 MB	33.980 MB	0.07847651	0.003163191377566
24	46.416 MB	–	100 MB	73.208 MB	0.07416942	0.006440862144930
25	100 MB	–	215.443 MB	157.721 MB	0.09371673	0.017533538933119
26	215.443 MB	–	464.159 MB	339.801 MB	0.08093624	0.032623369369260
27	464.159 MB	–	1 GB	732.079 MB	0.05399279	0.046887264039909
28	1 GB	–	2.154 GB	1.577 GB	0.04992384	0.093402916675691
29	2.154 GB	–	4.642 GB	3.398 GB	0.08871583	0.357591270942897
30	4.642 GB	–	10 GB	7.321 GB	0.03182476	0.276365775047813
31	10 GB	–	21.544 GB	15.772 GB	0.00452804	0.084715467164424
32	21.544 GB	–	46.416 GB	33.980 GB	0.00146156	0.058911819675084
33	46.416 GB	–	100 GB	73.208 GB	0.00017060	0.014814880370463
34	100 GB	–	215.443 GB	157.721 GB	0.00001375	0.002568882068470
35	215.443 GB	–	464.159 GB	339.801 GB	0.00000206	0.000829598407954
36	464.159 GB	–	1 TB	732.079 GB	0.00000069	0.000599130022577
37	1 TB	–	2.154 TB	1.577 TB	0.00000033	0.000616531696433
38	2.154 TB	–	4.310 TB	3.230 TB	0.00000001	0.000038314523896



(a) CERN file size distribution



(b) CERN sampled file size distribution

Figure 24. CERN file size distributions v_j and $\hat{v}_j$.

the mean $E(\hat{e}_s)$ is equal to 10.5 GB, the standard deviation is 53.1 GB, the second moment is 2,935 GB^2, and the coefficient of variation is equal to 5.05. By considering the file sizes $e_{s,j}$ to be the bin mean sizes, the mean $E(\hat{e}_s)$ is equal to 10.4 GB, the standard deviation is 52.6 GB, the second moment is 2,873 GB^2, and the coefficient of variation is equal to 5.03.

From the above discussion, and analogous to (112), it follows that the probability $\tilde{q}_{\hat{u}}$ that the single entity is lost is determined by

$$\tilde{q}_{\hat{u}} = \sum_{j=1}^{E_s} \tilde{q}_u(e_{s,j})\, \hat{v}_j \ . \tag{115}$$

Similarly, and analogous to (113), the expected amount $\breve{q}_{\hat{u}}$ of lost user data of the single entity is determined by

$$\breve{q}_{\hat{u}} = \sum_{j=1}^{E_s} \breve{q}_u(e_{s,j})\, \hat{v}_j \ . \tag{116}$$

Let Y_{U} be the number of lost entities and $\breve{Q}_{\mathrm{U}}$ the amount

of lost user data at exposure level u during the rebuild process of the C_u codewords. Then it holds that

$$E(Y_{\mathrm{U}}|C_u) \;=\; E_u\;\tilde{q}_u \;\overset{(108)}{\approx}\; \frac{C_u}{E(J)}\;\tilde{q}_u\;, \tag{117}$$

and

$$E(\breve{Q}_{\mathrm{U}}|C_u) \;=\; E_u\;\breve{q}_u \;\overset{(108)}{\approx}\; \frac{C_u}{E(J)}\;\breve{q}_u\;. \tag{118}$$

Note that $E(Y_{\mathrm{U}}|C_u)$, as determined by (117), can be obtained from Eq. (71) of [7] by replacing the shard size J with its average value $E(J)$. Consequently, (66) and (69) are obtained from the corresponding Eqs. (42) and (44) of [7] by replacing the shard size J with its average value $E(J)$.

Note also that $E(\breve{Q}_{\mathrm{U}}|C_u)$, as determined by (118), can be obtained from (117) by replacing the probability $\tilde{q}_u$ that an arbitrary entity is lost with its expected amount $\breve{q}_u$ of lost user data. Consequently, (86) is obtained from (66) by replacing $\tilde{q}_u$ with $\breve{q}_u$.

Remark 19: According to Remark 18, in the case where $C_u \ll E(J)$, it holds that $E_u = 1$, that is, a single entity is to be recovered. In this case, and considering (115), (116), (117), and (118), we have

$$E(Y_{\mathrm{U}}|C_u) \;=\; \tilde{q}_{\hat{u}}\;, \quad \text{for } C_u \ll E(J)\;. \tag{119}$$

and

$$E(\breve{Q}_{\mathrm{U}}|C_u) \;=\; \breve{q}_{\hat{u}}\;, \quad \text{for } C_u \ll E(J)\;. \tag{120}$$

According to (46), it holds that $C_u \;\approx\; C\,\prod_{i=1}^{u-1} V_i\,\alpha_i$. Consequently, condition $C_u \;\ll\; E(J)$ holds when the α_i variables take very small values. Note that, according to (45), these variables are approximately either equal to 1 or uniformly distributed in $(0, 1)$. Therefore, the region that corresponds to very small values of these variables is negligible. Consequently, Eqs. (66) and (69), which are obtained exclusively based on (117) and (118) without taking into consideration (119) and (120), are good approximations.

□

APPENDIX C

Proof of Corollary 2.

For a deterministic rebuild time distribution, it holds that $E(X^k) = [E(X)]^k$. Consequently, for $u = d+2, \ldots, \tilde{r}$, and from (75), it follows that

$$f_u \triangleq \frac{P^{(\tilde{r})}_{\ddot{s},u+1}}{P^{(\tilde{r})}_{\ddot{s},u}} = \frac{(\tilde{r}-u)(m-u+1)\,(u-d)\,\tilde{n}_u\,b_{u-1}\,V_u}{(\tilde{r}-u+1)(m-u)(u-d+1)\,\tilde{n}_{u-1}\,b_u}\;. \tag{121}$$

We shall now show that $f_u < 1$.

For the symmetric placement scheme, and using (36), (121) yields

$$\begin{aligned} f_u &= \frac{(\tilde{r}-u)(m-u+1)\,(u-d)\,(k-u)\frac{\min((k-u+1)\,b,B_{\max})}{l+1}\,\frac{m-u}{k-u}}{(\tilde{r}-u+1)(m-u)(u-d+1)\,(k-u+1)\frac{\min((k-u)\,b,B_{\max})}{l+1}} \\ &= \frac{(\tilde{r}-u)(m-u+1)\,(u-d)\;\min((k-u+1)\,b,B_{\max})}{(\tilde{r}-u+1)(k-u+1)(u-d+1)\;\min((k-u)\,b,B_{\max})} \\ &= \frac{\tilde{r}-u}{\tilde{r}-u+1}\;\frac{u-d}{u-d+1}\;\frac{m+1-u}{k-u}\;\frac{\min(b,\frac{B_{\max}}{k+1-u})}{\min(b,\frac{B_{\max}}{k-u})}\;. \end{aligned} \tag{122}$$

The fact that $\frac{B_{\max}}{k+1-u} < \frac{B_{\max}}{k-u}$ implies that $\min(b, \frac{B_{\max}}{k+1-u}) \leq \min(b, \frac{B_{\max}}{k-u})$ and therefore the last fraction is less than or equal to 1. Also, given that $k \geq m+1$, the third fraction is less than or equal to 1. Moreover, each of the first two fractions is less than 1. Consequently, $f_u < 1$.

For the clustered placement scheme, and using (42), (121) yields

$$\begin{aligned} f_u &= \frac{(\tilde{r}-u)(m-u+1)\,(u-d)\,(m-u)\;\min(b\,,\frac{B_{\max}}{l})}{(\tilde{r}-u+1)(m-u)(u-d+1)\,(m-u+1)\;\min(b\,,\frac{B_{\max}}{l})} \\ &= \frac{\tilde{r}-u}{\tilde{r}-u+1}\;\frac{u-d}{u-d+1}\;<\;1\;. \end{aligned} \tag{123}$$

□

APPENDIX D

Proof of Corollary 3.

For $u = d+2, \ldots, \tilde{r}$, and from (75), it follows that

$$g_u \triangleq \frac{P^{(\tilde{r})}_{\ddot{s},\tilde{r}}}{P^{(\tilde{r})}_{\ddot{s},u}} = \frac{(m-u+1)\,(u-d)\;\tilde{n}_{\tilde{r}-1}\,b_{u-1}}{(\tilde{r}-u+1)\,(m-\tilde{r}+1)\,(\tilde{r}-d)\;\tilde{n}_{u-1}\,b_{\tilde{r}-1}} \cdot \frac{E(X^{u-d-2})\,E(X^{\tilde{r}-d-1})}{E(X^{u-d-1})\,E(X^{\tilde{r}-d-2})} \cdot \prod_{i=u}^{\tilde{r}-1} V_i\;. \tag{124}$$

For a Weibull rebuild time distribution, with probability density and cumulative distribution functions

$$\begin{aligned} f_X(x;\eta,\Lambda) \;&=\; \frac{\eta}{\Lambda}\left(\frac{x}{\Lambda}\right)^{\eta-1} e^{-(x/\Lambda)^\eta} \\ F_X(x;\eta,\Lambda) \;&=\; 1 - e^{-(x/\Lambda)^\eta}\;, \end{aligned} \tag{125}$$

it holds that

$$E(X^k) = \Lambda^k\,\Gamma(1+k/\eta)\;. \tag{126}$$

Note that this distribution provides a continuous spectrum between the deterministic distribution (for $\eta \to \infty$) and the exponential distribution (for $\eta = 1$). Let us introduce the variable h_u defined as follows:

$$h_u \triangleq \frac{E(X^{u-d-2})\,E(X^{\tilde{r}-d-1})}{E(X^{u-d-1})\,E(X^{\tilde{r}-d-2})}\;. \tag{127}$$

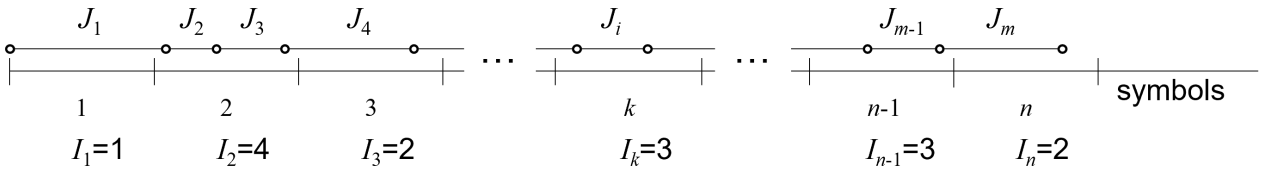


Figure 25. Number of shards that symbols span.

By virtue of (126), (127) yields

$$h_u \triangleq \frac{\Lambda^{u-d-2}\,\Gamma\left(1+\frac{u-d-2}{\eta}\right)\Lambda^{\tilde{r}-d-1}\,\Gamma\left(1+\frac{\tilde{r}-d-1}{\eta}\right)}{\Lambda^{u-d-1}\,\Gamma\left(1+\frac{u-d-1}{\eta}\right)\Lambda^{\tilde{r}-d-2}\,\Gamma\left(1+\frac{\tilde{r}-d-2}{\eta}\right)} = \frac{\Gamma\left(1+\frac{u-d-2}{\eta}\right)\Gamma\left(1+\frac{\tilde{r}-d-1}{\eta}\right)}{\Gamma\left(1+\frac{u-d-1}{\eta}\right)\Gamma\left(1+\frac{\tilde{r}-d-2}{\eta}\right)} . \quad (128)$$

From (126) and for $n \to \infty$, it holds that $E(X^k) = \Lambda^k = [E(X)]^k$ and consequently $h_u = 1$. For $n = 1$, it holds that $E(X^k) = \Lambda^k\,\Gamma(1+k) = k!\,\Lambda^k = k!\,[E(X)]^k$ and consequently $h_u = (\tilde{r}-d-1)/(u-d-1)$. As the function $\Gamma(x)$ is convex, it holds that h_u decreases with increasing η, such that

$$1 \le h_u \le \frac{\tilde{r}-d-1}{u-d-1} , \quad \text{for } 1 \le \eta < \infty . \quad (129)$$

For the symmetric placement scheme, and using (37) and the fact that $k \ge m+1$, it holds that

$$\prod_{i=n_1}^{n_2} V_i = \prod_{i=n_1}^{n_2} \frac{m-u}{k-u} < \prod_{i=n_1}^{n_2} \frac{m-u}{m+1-u} = \frac{m-n_2}{m+1-n_1} . \quad (130)$$

For the symmetric placement scheme, and using (35), (36) and (127), (124) yields

$$g_u = \frac{(m-u+1)\,(u-d)\,(k-\tilde{r}+1)\,\frac{\min((k-u+1)\,b,B_{\max})}{l+1}}{(\tilde{r}-u+1)(m-\tilde{r}+1)\,(\tilde{r}-d)\,(k-u+1)\,\frac{\min((k-\tilde{r}+1)\,b,B_{\max})}{l+1}} \cdot \frac{E(X^{u-2})\,E(X^{\tilde{r}-1})}{E(X^{u-1})\,E(X^{\tilde{r}-2})} \cdot \prod_{i=u}^{\tilde{r}-1} V_i = \frac{(m-u+1)\,(u-d)\,\min(b,\frac{B_{\max}}{k-u+1})}{(\tilde{r}-u+1)(m-\tilde{r}+1)\,(\tilde{r}-d)\,\min(b,\frac{B_{\max}}{k-\tilde{r}+1})}\; h_u \prod_{i=u}^{\tilde{r}-1} V_i . \quad (131)$$

Given that $u \le \tilde{r}-1 < \tilde{r}$, it holds that $\frac{B_{\max}}{k-u+1} < \frac{B_{\max}}{k-\tilde{r}+1}$ and therefore $\min(b,\frac{B_{\max}}{k-u+1}) < \min(b,\frac{B_{\max}}{k-\tilde{r}+1})$. Consequently, using (129) and (130), (131) yields

$$g_u < \frac{(m-u+1)\,(u-d)}{(\tilde{r}-u+1)(m-\tilde{r}+1)\,(\tilde{r}-d)}\;\frac{\tilde{r}-d-1}{u-d-1}\;\frac{m-\tilde{r}+1}{m+1-u} = \frac{u-d}{\tilde{r}-d}\;\frac{\tilde{r}-d-1}{(\tilde{r}-u+1)(u-d-1)} < \frac{\tilde{r}-d-1}{(\tilde{r}-u+1)(u-d-1)} . \quad (132)$$

Given that $d+2 \le u < \tilde{r}$, it holds that $[u-(d+2)](u-\tilde{r}) \le 0$ or, equivalently, $\tilde{r}-d-1 \le (\tilde{r}-u+1)(u-d-1)$. Consequently, it follows from (132) that $g_u < 1$.

For the clustered placement scheme, and using (41), (42), (43), and (127), (131) yields

$$g_u = \frac{(m-u+1)\,(u-d)\,(m-\tilde{r}+1)\,h_u\,\min(b\,,\frac{B_{\max}}{l})}{(\tilde{r}-u+1)(m-\tilde{r}+1)\,(\tilde{r}-d)\,(m-u+1)\,\min(b\,,\frac{B_{\max}}{l})} \overset{(129)}{\le} \frac{u-d}{\tilde{r}-d}\;\frac{\tilde{r}-d-1}{(\tilde{r}-u+1)(u-d-1)} < \frac{\tilde{r}-d-1}{(\tilde{r}-u+1)(u-d-1)} . \quad (133)$$

As the last term of (133) is the same as the last term of (132), which is less or equal to 1, it follows that $g_u < 1$.

□

APPENDIX E

Proof of Corollary 4.

Immediate from Corrolary 1, (16), (66), and (72).

Relation (83) can alternatively be obtained as follows. At exposure level u and for very small values of P_s, an entity failure is most likely caused by a single corrupted codeword that loses $\tilde{r}$ symbols. Let I be the number of shards that have parts stored in a symbol of this codeword. Then the expected number $E(Y_{\text{UF}_u})$ of lost entities associated with the direct path $\overrightarrow{UF_u}$ is determined by

$$E(Y_{\text{UF}_u}) \approx E(I)\; P_{\text{UF}_u} , \quad (134)$$

where P_{UF_u} is the probability of data loss due to unrecoverable symbol errors at exposure level u.

We proceed to show that

$$E(I) = 1 + \frac{1}{E(J)} . \quad (135)$$

Let us consider m successive shards stored in n symbols, as depicted in Figure 25, with the shard boundaries indicated by the circles and the symbol boundaries indicated by the vertical lines. Let J_i denote their size ($i = 1, \ldots, m$) and let I_k ($k = 1, \ldots, n$) denote the number of shards that have parts stored in the k-th symbol. For large values of m and n, it holds that

$$\sum_{i=1}^{m} J_i \approx n , \quad (136)$$

such that

$$\lim_{m\to\infty} \frac{\sum_{i=1}^{m} J_i}{n} = 1 . \quad (137)$$

It also holds that

$$E(J) = \frac{\sum_{i=1}^{m} J_i}{m} , \quad (138)$$

and

$$E(I) = \frac{\sum_{k=1}^{n} I_k}{n} \ , \tag{139}$$

Combining (137) and (138) yields

$$\lim_{m\to\infty} \frac{m}{n} = \frac{1}{E(J)} \ . \tag{140}$$

Note that the number of shards that have parts stored in a symbol decreased by one is equal to the number of shard boundaries within the symbol. For instance, regarding the kth symbol, there are three shards that have parts stored in this symbol, namely the $(i-1)$th, ith, and $(i+1)$th shard, such that $I_k = 3$, which decreased by one yields the two shard boundaries within this symbol. Consequently, considering the n symbols and the corresponding m boundaries, we have

$$\sum_{k=1}^{n} (I_k - 1) = m \tag{141}$$

or

$$\sum_{k=1}^{n} I_k = n + m \tag{142}$$

Substituting (142) into (139), and using (140), yields (135).

An alternative proof for the case where $J_j \geq 1$, for $j = 1, 2, \ldots, E_s$, is the following. Let us consider an arbitrary symbol and let $\hat{J}$ be the size of the shard that is stored at the beginning of the symbol. As this shard is a sampled shard, the pdf of its size $\hat{J}$ is determined by (114), that is,

$$P(\hat{J} = J_j) = P(\hat{e}_s = e_{s,j}) = \hat{v}_j \ , \quad \text{for } \ j = 1, 2, \ldots, E_s \ . \tag{143}$$

Let y be the size from the beginning of the sampled shard to the beginning of the symbol. Then y is uniformly distributed in the interval $(0, \hat{J})$. For y in the interval $(0, \hat{J}-1)$, the symbol only contains a part of the sampled shard, that is, it contains a part of a single shard. The probability p of this event is

$$p = \int_0^{\hat{J}-1} \frac{1}{\hat{J}} \, dx = \frac{\hat{J}-1}{\hat{J}} \ . \tag{144}$$

On the other hand, for y in the interval $(\hat{J}-1, 1)$, the symbol contains parts of the sampled shard, as well as of the subsequent shard, that is, the symbol contains parts of two shards. The probability of this event is $1-p$. Consequently, the expected number $E(I|\hat{J})$ of shards that have parts stored in the symbol is

$$E(I|\hat{J}) = 1 \cdot p + 2 \cdot (1-p) = 2 - p \overset{(144)}{=} \frac{\hat{J}+1}{\hat{J}} \ . \tag{145}$$

Unconditioning (145) on $\hat{J}$ and using (143) yields

$$E(I) = \sum_{j=1}^{E_s} E(I|J_j) \, P(\hat{J} = J_j) \overset{(145)}{=} \sum_{j=1}^{E_s} \frac{J_j + 1}{J_j} \, \hat{v}_j$$
$$\overset{(14)(16)(114)}{=} \sum_{j=1}^{E_s} \frac{J_j + 1}{J_j} \cdot \frac{J_j \, v_j}{E(J)} = \frac{E(J)+1}{E(J)} = 1 + \frac{1}{E(J)} \ , \tag{146}$$

which is relation (135).

Substituting (135) into (134) yields (83).

□

REFERENCES

[1] I. Iliadis, "Relations between entity sizes and error-correction coding codewords and data loss," in Proceedings of the 17th International Conference on Communication Theory, Reliability, and Quality of Service (CTRQ), May 2024, pp. 1–11.

[2] I. Iliadis and V. Venkatesan, "Reliability evaluation of erasure coded systems," Int'l J. Adv. Telecommun., vol. 10, no. 3&4, Dec. 2017, pp. 118–144.

[3] I. Iliadis, "Reliability evaluation of erasure coded systems under rebuild bandwidth constraints," Int'l J. Adv. Networks and Services, vol. 11, no. 3&4, Dec. 2018, pp. 113–142.

[4] ——, "Reliability of erasure-coded storage systems with latent errors," Int'l J. Adv. Telecommun., vol. 15, no. 3&4, Dec. 2022, pp. 23–41.

[5] ——, "Reliability evaluation of erasure-coded storage systems with latent errors," ACM Trans. Storage, vol. 19, no. 1, Jan. 2023, pp. 1–47. [Online]. Available: https://doi.org/10.1145/3568313

[6] I. Iliadis and V. Venkatesan, "Expected annual fraction of data loss as a metric for data storage reliability," in Proceedings of the 22nd Annual IEEE International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS), Sep. 2014, pp. 375–384.

[7] I. Iliadis, "Expected annual fraction of entity loss as a metric for data storage durability," in Proceedings of the 16th International Conference on Communication Theory, Reliability, and Quality of Service (CTRQ), Apr. 2023, pp. 1–11.

[8] G. A. Jaquette, "LTO: A better format for mid-range tape," IBM J. Res. Dev., vol. 47, no. 4, Jul. 2003, pp. 429–444.

[9] I. Iliadis, R. Haas, X.-Y. Hu, and E. Eleftheriou, "Disk scrubbing versus intradisk redundancy for RAID storage systems," ACM Trans. Storage, vol. 7, no. 2, Jul. 2011, pp. 1–42.

[10] I. Iliadis and V. Venkatesan, "Most probable paths to data loss: An efficient method for reliability evaluation of data storage systems," Int'l J. Adv. Syst. Measur., vol. 8, no. 3&4, Dec. 2015, pp. 178–200.

[11] A. Dholakia, E. Eleftheriou, X.-Y. Hu, I. Iliadis, J. Menon, and K. Rao, "A new intra-disk redundancy scheme for high-reliability RAID storage systems in the presence of unrecoverable errors," ACM Trans. Storage, vol. 4, no. 1, May 2008, pp. 1–42.

[12] I. Iliadis, "Reliability modeling of RAID storage systems with latent errors," in Proceedings of the 17th Annual IEEE/ACM International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS), Sep. 2009, pp. 111–122.

[13] T. J. E. Schwarz, Q. Xin, E. L. Miller, D. D. E. Long, A. Hospodor, and S. Ng, "Disk scrubbing in large archival storage systems," in Proceedings of the 12th Annual IEEE/ACM International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS), Oct. 2004, pp. 409–418.

[14] A. Oprea and A. Juels, "A clean-slate look at disk scrubbing," in Proceedings of the 8th USENIX Conference on File and Storage Technologies (FAST), Feb. 2010, pp. 57–70.

[15] B. Schroeder, S. Damouras, and P. Gill, "Understanding latent sector errors and how to protect against them," ACM Trans. Storage, vol. 6, no. 3, Sep. 2010, pp. 1–23.

[16] S. A. Chamazcoti, B. Safaei, and S. G. Miremadi, "Can erasure codes damage reliability in ssd-based storage systems?" IEEE Transactions on Emerging Topics in Computing, vol. 7, no. 3, 2019, pp. 435–446.

[17] M. Kishani, S. Ahmadian, and H. Asadi, "A modeling framework for reliability of erasure codes in ssd arrays," IEEE Transactions on Computers, vol. 69, no. 5, 2020, pp. 649–665.

[18] M. Zhang, S. Han, and P. P. C. Lee, "SimEDC: A simulator for the reliability analysis of erasure-coded data centers," IEEE Trans. Parallel Distrib. Syst., vol. 30, no. 12, 2019, pp. 2836–2848.

[19] M. Silberstein, L. Ganesh, Y. Wang, L. Alvisi, and M. Dahlin, "Lazy means smart: Reducing repair bandwidth costs in erasure-coded distributed storage," in Proceedings of the 7th ACM International Systems and Storage Conference (SYSTOR), Jun. 2014, pp. 15:1–15:7.

[20] Tape Roadmap, Information Storage Industry Consortium (INSIC) Report, 2019. [Online]. Available: https://www.insic.org/wp-content/uploads/2019/07/INSIC-Applications-and-Systems-Roadmap.pdf [retrieved: November, 2024]

[21] I. Iliadis, "Effect of lazy rebuild on reliability of erasure-coded storage systems," in Proceedings of the 15th International Conference on Communication Theory, Reliability, and Quality of Service (CTRQ), Apr. 2022, pp. 1–10.

[22] G. Cancio et al., "Tape archive challenges when approaching exabyte-scale," 2010, Presentation at CHEP 2010, available online.

[23] I. Iliadis, L. Jordan, M. Lantz, and S. Sarafijanovic, "Performance evaluation of automated tape library systems," in Proceedings of the 29th IEEE International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS), Nov. 2021, pp. 1–8.

[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 Proceedings of the 9th USENIX Symposium on Operating Systems Design and Implementation (OSDI), Oct. 2010, pp. 61–74.

[25] M. Ovsiannikov, S. Rus, D. Reeves, P. Sutter, S. Rao, and J. Kelly, "The quantcast file system," in Proceedings of the 39th International Conference on Very Large Data Bases (VLDB), vol. 6, no. 11. VLDB Endowment, Aug. 2013, pp. 1092–1101.

[26] C. Huang, H. Simitci, Y. Xu, A. Ogus, B. Calder, P. Gopalan, J. Li, and S. Yekhanin, "Erasure coding in Windows Azure Storage," in Proceedings of the USENIX Annual Technical Conference (ATC), Jun. 2012, pp. 15–26.

[27] S. Muralidhar et al., "f4: Facebook's Warm BLOB Storage System," in Proceedings of the 11th USENIX Symposium on Operating Systems Design and Implementation (OSDI), Oct. 2014, pp. 383–397.

[28] Amazon Simple Storage Service (Amazon S3), 2024. [Online]. Available: http://aws.amazon.com/s3/ [retrieved: November, 2024]

[29] I. Iliadis and M. Lantz, "Reliability evaluation of automated tape library systems," in Proceedings of the 32nd IEEE International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS), Oct. 2024, pp. 80–87.

[30] L. Kleinrock, Queueing Systems, Volume 1: Theory. New York: Wiley, 1975.