Relations Between Entity Sizes and Error-Correction Coding Codewords and 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 EAFEL and EAFEDL metrics are assessed analytically for erasure-coding redundancy schemes and for the clustered, declustered, and symmetric data placement schemes. 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.**

Keywords–Reliability analysis; MTTDL; EAFDL; EAFEL; MDS codes; Unrecoverable or latent sector 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. Desired reliability levels are ensured by employing erasure-coding redundancy schemes for recovering lost data [1-4].

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 [3][4]. Also, the amount of data loss is obtained by the Expected Annual Fraction of Data Loss (EAFDL) metric that was introduced in [5]. This metric was recently complemented by the Expected Annual Fraction of Entity Loss (EAFEL) metric [6]. 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 [7]. 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 [6]. 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 [6] 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 [6] 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, is ex-

Copyright (c) IARIA, 2024. ISBN: 978-1-68558-172-5

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 size
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 $(C = c/s)$
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

tended to derive analytically the EAFEL and the new EAFEDL reliability metrics for the case of variable-size entities. Subsequently, we demonstrate the effect of erasure-coding capability as well as of entity and symbol sizes on system reliability for the entire range of bit error rates.

The remainder of the article is organized as follows. Section II describes the storage system model and the corresponding parameters considered. In Section III, 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 IV, the EAFEL and EAFEDL metrics are derived analytically for the case of random placement of variable-size entities. Section V 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. Finally, we conclude in Section VI.

II. STORAGE SYSTEM MODEL

The reliability of erasure-coded storage systems was assessed in [6] 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. Also, 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 way, the system can tolerate any $\tilde{r} - 1$ device failures, but $\tilde{r}$ device failures may lead to data loss, with

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

Two different ways (A and B) for storing user data on devices were shown in Figure 1 of [6]. 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 $s_s = e_s/l$.

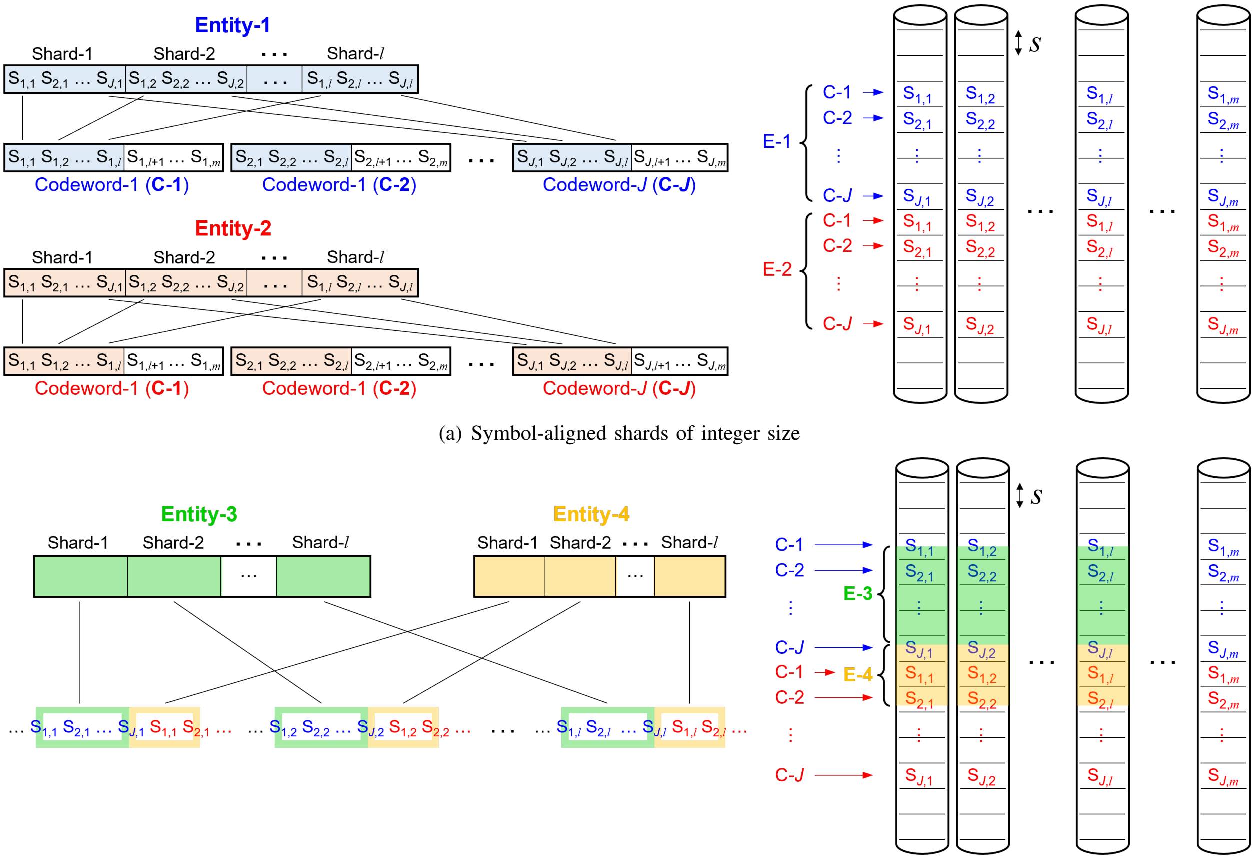
The storage space of devices is partitioned into units (symbols) of a fixed size s and complemented with parity symbols to form codewords. Each shard was assumed to be stored in an integer number of J symbols that is determined by

$$J = \frac{s_s}{s} = \frac{e_s}{l\, s} \,. \qquad (4)$$

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 [6] 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.

Copyright (c) IARIA, 2024. ISBN: 978-1-68558-172-5


Entity-1
Shard-1
Shard-2
Shard-l
$S_{1,1}$ $S_{2,1}$ … $S_{J,1}$
$S_{1,2}$ $S_{2,2}$ … $S_{J,2}$
$S_{1,l}$ $S_{2,l}$ … $S_{J,l}$
$S_{1,1}$ $S_{1,2}$ … $S_{1,l}$ $S_{1,l+1}$ … $S_{1,m}$
$S_{2,1}$ $S_{2,2}$ … $S_{2,l}$ $S_{2,l+1}$ … $S_{2,m}$
$S_{J,1}$ $S_{J,2}$ … $S_{J,l}$ $S_{J,l+1}$ … $S_{J,m}$
Codeword-1 (C-1)
Codeword-1 (C-2)
Codeword-J (C-J)
Entity-2
E-1
E-2
C-1
C-2
C-J
s
Entity-3
Entity-4
E-3
E-4
$S_{1,m}$
$S_{2,m}$
$S_{J,m}$

(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.

As pointed out in [6], 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 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 III.

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 $P_{\mathrm{b}} \approx 10^{-12}$ [4]. On the other hand, according to Figure 13 in [8], 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. Moreover, latent errors are found to exhibit spatial locality and they occur in bursts of multiple contiguous symbol errors. The degree to which symbol errors are correlated is captured by the factor f_{cor} whose value is typically close to 1 [4].

III. 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 $[\epsilon, S + \epsilon]$ with $0 < \epsilon < 1$. 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 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.

Owing to periodicity, it suffices to study the process in the symbol interval $[\epsilon, 23 + \epsilon]$. The red integers indicate the

Copyright (c) IARIA, 2024. ISBN: 978-1-68558-172-5

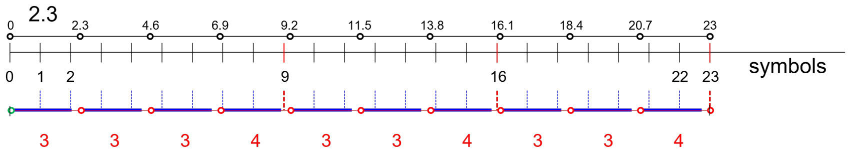


Figure 2. Number of symbols that shards span. Fixed-size shards of size 2.3 symbols.

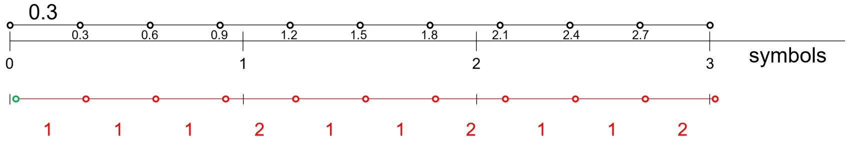


Figure 3. Number of symbols that shards span. Fixed-size shards of size 0.3 symbols.

number of symbols spanned by the successive shards. We note that 7 shards span 3 symbol and the remaining 3 shards span 4 symbols. Therefore, the probability density function (pdf) $\{p_j\}$ of the number of symbols K that an arbitrary shard spans is

$$P(K=i) = p_i = \begin{cases} 0.7\,, & \text{for} \quad i=3 \\ 0.3\,, & \text{for} \quad i=4\,. \end{cases} \tag{6}$$

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

$$P(K=i) = p_i = \begin{cases} 1 - fr(J)\,, & \text{for} \quad i = \lfloor J \rfloor + 1 \\ fr(J)\,, & \text{for} \quad i = \lfloor J \rfloor + 2 \\ 0\,, & \text{otherwise}\,, \end{cases} \tag{7}$$

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

$$fr(x) \triangleq x - \lfloor x \rfloor\,, \quad \forall x \in \mathcal{R}\,. \tag{8}$$

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. Therefore, the pdf $\{p_j\}$ of the number of codewords (symbols) K that an arbitrary entity (shard) spans is

$$P(K=i) = p_i = \begin{cases} 0.7\,, & \text{for} \quad i=1 \\ 0.3\,, & \text{for} \quad i=2\,, \end{cases} \tag{9}$$

which is also the result determined by (7).

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 (7), the pdf $\{p_j\}$ of the number of codewords (symbols) K that an arbitrary entity (shard) spans is

$$P(K=i) = p_i = \begin{cases} 0.3\,, & \text{for} \quad i=3 \\ 0.7\,, & \text{for} \quad i=4\,, \end{cases} \tag{10}$$

which is also the result determined by (7).

B. Variable-Size Entities

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

$$v_j \triangleq P(e_s = e_{s,j})\,, \quad \text{for} \quad j = 1, 2, \ldots, L\,, \tag{11}$$

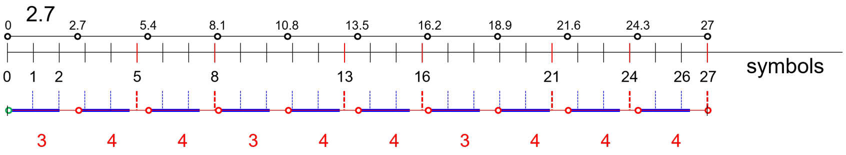
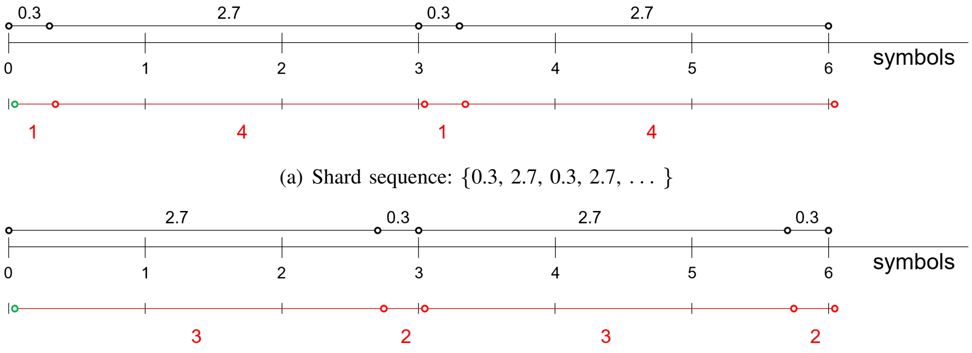


Figure 4. Number of symbols that shards span. Fixed-size shards of size 2.7 symbols.



(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 $v_1 = v_2 = 0.5$.

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

$$E(e_s) = \sum_{j=1}^{L} e_{s,j}\, v_j \ . \tag{12}$$

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

$$J_j = \frac{e_{s,j}}{l\, s} \quad \text{for} \ \ j = 1, 2, \ldots, L \ . \tag{13}$$

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

$$P(J = J_j) = v_j \ , \quad \text{for} \ \ j = 1, 2, \ldots, L \ , \tag{14}$$

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

$$E(J) \ = \ \sum_{j=1}^{L} J_j\, v_j \ \overset{(12)(13)}{=} \ \frac{E(e_s)}{l\, s} \ . \tag{15}$$

The preceding discussion begs the following questions. Can the probability density function $\{p_j\}$ that was theoretically obtained in (7) for the case of a single fixed shared 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 (7) and (14) we deduce that

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

$$\text{for } j = 1, 2, \ldots, L \ . \tag{16}$$

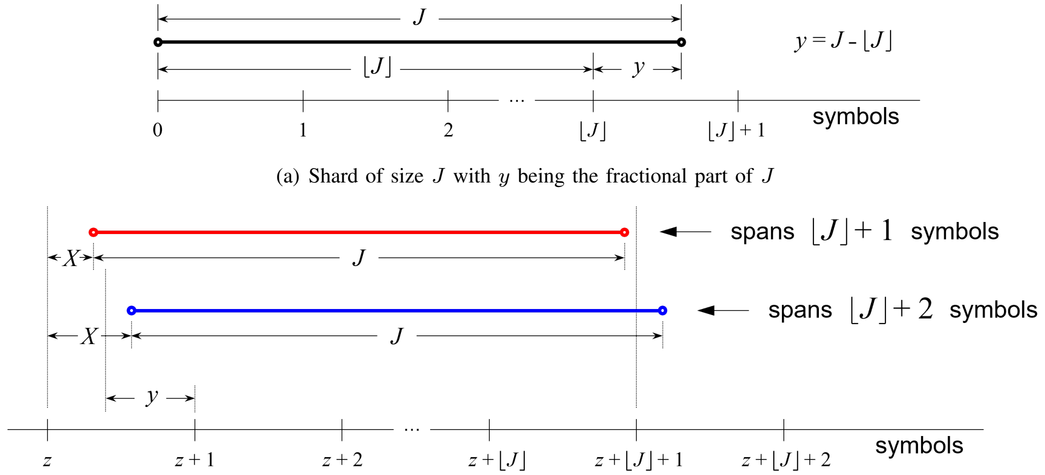
Let us consider the special case of a discrete bimodal distribution for the shard size, that is, $L = 2$, and let us assume that half of the shards have a size of 0.3 symbols and the remaining half of the shards have a size of 2.7 symbols. In this case we have $J_1 = 0.3$, $J_2 = 2.7$, and $v_1 = v_2 = 0.5$. For the particular realization where first the shards of size 0.3 are stored followed by the shards of size 2.7, (16) yields

$$P(K=i) = p_i = \begin{cases} 0.7 \times 0.5 = 0.35 \, , & \text{for} \ \ i = 1 \\ 0.3 \times 0.5 = 0.15 \, , & \text{for} \ \ i = 2 \\ 0.3 \times 0.5 = 0.15 \, , & \text{for} \ \ i = 3 \\ 0.7 \times 0.5 = 0.35 \, , & \text{for} \ \ i = 4 \\ 0 \ , & \text{otherwise.} \end{cases} \tag{17}$$

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 $v_j = 1/L$, for $j = 1, 2, \ldots, L$, 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_L 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 III-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 $[\epsilon, 3+\epsilon]$. The red integers indicate the number of symbols spaned by the successive shards. We note that half of the shards span 1 symbol and the remaining half of the shards span 4 symbols. Consequently, the pdf $\{p_j\}$ of the number of symbols K that



(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.

an arbitrary shard spans is

$$P(K=i)=p_i=\begin{cases}0.5\,, & \text{for } \ i=1\\ 0.5\,, & \text{for } \ i=4\,.\end{cases} \tag{18}$$

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). In this case, half of the shards span 3 symbols and the remaining half of the shards span 2 symbols. Consequently, the pdf $\{p_j\}$ of the number of symbols K that an arbitrary shard spans is

$$P(K=i)=p_i=\begin{cases}0.5\,, & \text{for } \ i=2\\ 0.5\,, & \text{for } \ i=3\,.\end{cases} \tag{19}$$

We now observe that the pdf determined by (19) is different from that determined by (18). Moreover, both of them, are different from that determined by (17) 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 (14), but also on their actual placement.

3) Random Shard Placement: According to this placement, successive shard sizes are assumed to be independent and identically distributed (i.i.d) according to the distribution given in (14). Let us consider a randomly chosen shard. Let also J denote its size, as shown in in Figure 6(a), and y its fractional part, that is, $y = J - \lfloor J \rfloor$. A randomly placed such shard 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 $X \le 1 - y$, the shard spans $\lfloor J \rfloor + 1$ symbols where as 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

$$P(K=\lfloor J \rfloor+1)=\int_0^{1-y} dx = 1-y\;, \tag{20}$$

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

$$P(K=\lfloor J \rfloor+2)=1-P(K=\lfloor J \rfloor+1) \overset{(20)}{=} y\;. \tag{21}$$

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

$$P(K=i)=p_i=\begin{cases}1-fr(J)\,, & \text{for } \ i=\lfloor J \rfloor+1\\ fr(J)\,, & \text{for } \ i=\lfloor J \rfloor+2\\ 0\;, & \text{otherwise}\,.\end{cases} \tag{22}$$

From (22), and using (8), it follows that the mean number $E(K)$ of symbols that a shard of size J spans is

$$\begin{aligned}E(K) &= (\lfloor J \rfloor+1)P(K=\lfloor J \rfloor+1)+(\lfloor J \rfloor+2)P(K=\lfloor J \rfloor+2)\\ &= (\lfloor J \rfloor+1)\,[1-fr(J)]+(\lfloor J \rfloor+2)\,fr(J)=J+1\,.\end{aligned} \tag{23}$$

From (14), (22), and (23), it follows that the pdf and the average number of symbols K that an arbitrary shard spans are determined by

$$P(K=i)=p_i=\begin{cases}[1-fr(J_j)]\,v_j\,, & \text{for } \ i=\lfloor J_j \rfloor+1\\ fr(J_j)\,v_j\,, & \text{for } \ i=\lfloor J_j \rfloor+2\\ 0\;, & \text{otherwise,}\end{cases}$$
$$\text{for } j=1,2,\ldots,L\;, \tag{24}$$

and

$$E(K)=\sum_{j=1}^{L}(\,J_j+1)\,v_j = E(J)+1\;. \tag{25}$$

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

Remark 2: From (16) and (24), 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.

IV. Derivation of EAFEL and EAFEDL

The EAFEL and EAFEDL reliability metrics are derived using the general methodology presented in [1-5], which we briefly review here. 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) or back to the original normal mode by restoring initial redundancy.

The EAFEL metric is obtained by Equation (16) of [6] as follows:

$$\text{EAFEL} \approx \frac{E(Y)}{E(T) \cdot N_E} \ , \tag{26}$$

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, of a typical interval of normal operation until the rebuild process of failed devices is triggered, which is determined by Equation (14) of [6]. The number N_E of entities in the system is

$$N_E \approx \frac{U}{E(e_s)} \overset{(1)}{=} \frac{n}{m} \cdot \frac{l\,c}{E(e_s)} \overset{(15)}{=} \frac{n}{m} \cdot \frac{c}{E(J)\,s} \ . \tag{27}$$

Similarly to Equation (9) of [5], 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:

$$\text{EAFEDL} \approx \frac{E(\breve{Q})}{E(T) \cdot U} \overset{(1)}{=} \frac{m\,E(\breve{Q})}{n\,l\,c\,E(T)} \ . \tag{28}$$

A. Reliability Analysis

The EAFEDL is evaluated in parallel with EAFEL using the theoretical framework presented in [6]. The system is at exposure level u $(0 \le u \le \tilde{r})$ 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, $(u \ge 1)$. Upon the first device failure it holds that

$$C_1 = C \ , \tag{29}$$

where C is determined by (2).

The reliability metrics of interest are derived using the *direct path approximation*, which considers only transitions from lower to higher exposure levels [1-5]. 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 $1 \to 2 \to \cdots \to \tilde{r}-1 \to \tilde{r}$.

1) 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 $\overrightarrow{\text{DF}}$ and $\overrightarrow{UF_u}$, respectively. Then, it holds that [6, Equations (37), (38), (41)]

$$E(Y) \approx E(Y_{\text{DF}}) + \sum_{u=d+1}^{\tilde{r}-1} E(Y_{\text{UF}_u}) \approx E(Y_{\text{DF}}) + E(Y_{\text{UF}}) \ , \tag{30}$$

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

$$E(Y_{\text{UF}}) \approx \sum_{u=1}^{\tilde{r}-1} E(Y_{\text{UF}_u}) \ . \tag{31}$$

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

$$E(Y_{\text{UF}_u}) \approx \frac{C}{E(J)} \, \frac{P_u}{u-d} \left(\prod_{j=1}^{u-1} V_j \right) \tilde{q}_u \ , \tag{32}$$

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

$$\tilde{q}_u = \sum_{j=1}^{L} \tilde{q}_{s,u} \left(\frac{e_{s,j}}{l\,s} \right) v_j \ , \tag{33}$$

with

$$\tilde{q}_{s,u}(x) \triangleq 1 - [1 - fr(x)]\, q_u^{f_{\text{cor}}\,(\lfloor x \rfloor + 1)} - fr(x)\, q_u^{f_{\text{cor}}\,(\lfloor x \rfloor + 2)} \ , \tag{34}$$

and q_u, which denotes the probability that a codeword that has lost u symbols can be restored, is determined by

$$q_u = 1 - \sum_{j=\tilde{r}-u}^{m-u} \binom{m-u}{j} P_s^j (1 - P_s)^{m-u-j} \ . \tag{35}$$

It also holds that

$$E(Y_{\text{DF}}) \approx \frac{C}{E(J)} \, \frac{P_{\text{DF}}}{\tilde{r}-d} \prod_{j=1}^{\tilde{r}-1} V_j \ , \tag{36}$$

where C is determined by (2), P_s is determined by (5), $fr(x)$ is determined by (8), $E(J)$ is determined by (15), and P_u, P_{DF}, and V_j are determined by Equations (29), (23), and (60) of [6], respectively.

Proof: Equation (32) is obtained in Appendix. Equation (36) is obtained from (32) 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 $P_{\tilde{r}} = P_{\text{DF}}$. ■

2) 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.

Similarly to (30), it holds that

$$E(\breve{Q}) \approx E(\breve{Q}_{\text{DF}}) + \sum_{u=d+1}^{\tilde{r}-1} E(\breve{Q}_{\text{UF}_u}) \approx E(\breve{Q}_{\text{DF}}) + E(\breve{Q}_{\text{UF}}) \ , \tag{37}$$

TABLE II. PARAMETER VALUES

Parameter	Definition	Values
n	number of storage devices	64
c	amount of data stored on each device	20 TB
s	symbol (sector) 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 ($0 \le d < m - l$)	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

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

$$E(\breve{Q}_{\text{UF}}) \approx \sum_{u=1}^{\tilde{r}-1} E(\breve{Q}_{\text{UF}_u}) \, . \tag{38}$$

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

$$E(\breve{Q}_{\text{UF}_u}) \;\approx\; \frac{C}{E(J)} \, \frac{P_u}{u-d} \left(\prod_{j=1}^{u-1} V_j \right) \breve{q}_u \;, \tag{39}$$

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

$$\breve{q}_u \;=\; \sum_{j=1}^{L} e_{s,j} \, \tilde{q}_{s,u} \left(\frac{e_{s,j}}{l\,s} \right) v_j \;. \tag{40}$$

It also holds that

$$E(\breve{Q}_{\text{DF}}) \;\approx\; \frac{C}{E(J)} \, \frac{P_{\text{DF}}}{\tilde{r}-d} \left(\prod_{j=1}^{\tilde{r}-1} V_j \right) \breve{q}_{\tilde{r}} \;, \tag{41}$$

where C is determined by (2), $E(J)$ is determined by (15), $\tilde{q}_{s,u}(x)$ is determined by (34), and P_u, P_{DF}, and V_j are determined by Equations (29), (23), and (60) of [6], respectively.

Proof: Equation (39) is obtained in Appendix. Equation (41) is obtained from (39) by setting $u = \tilde{r}$ and recognizing that $P_{\tilde{r}} = P_{\text{DF}}$. From (40), (12), it follows that $\breve{q}_{\tilde{r}} = E(e_s)$. ■

V. NUMERICAL RESULTS

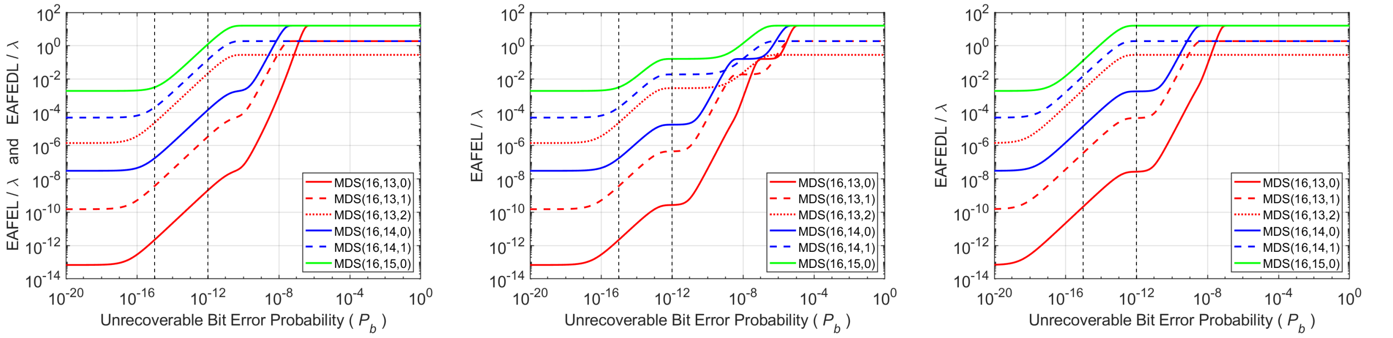
Here, we assess the reliability of the clustered and declustered placement schemes for the system and the parameter values considered in [6], as listed in Table II. The system is comprised of $n = 64$ devices (HDDs), it is protected by MDS erasure codes with $m = 16$ and $l = 13$, 14, 15 and employs 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 parameter λ^{-1} is chosen to be equal to $876,000$ 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 $\mu^{-1} = c/b = 55.5$ h, such that $\lambda/\mu = 6.3 \times 10^{-5} \ll 1$, which is a condition that ensures the accuracy of the reliability results obtained. Also, the rebuild time distribution is deterministic and sector errors are correlated with $f_{\text{cor}} \approx 1$.

First, we assess the reliability for the declustered placement scheme ($k = n = 64$) for the MDS-coded configurations considered in [6] 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 7 and 8 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 EAFEL/λ reliability metric corresponding to the declustered data placement scheme is obtained from (26) and shown in Figure 7(a) for a fixed entity size of $e_s = 10$ GB. In the interval $[10^{-15}, 10^{-12}]$ of practical importance for P_{b}, which is indicated between the two vertical dashed lines, EAFEL is degraded by orders of magnitude. Note that in the case of fixed-size entities, 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, the same as the entity size e_s in the fixed-entity-size case considered previously. From (15), it follows that the average shard size $E(J)$ remains the same, which, according to (27), 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 7(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 7(c) that shows the EAFEDL metric. Note that for $P_{\text{b}} = 10^{-15}$, 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.

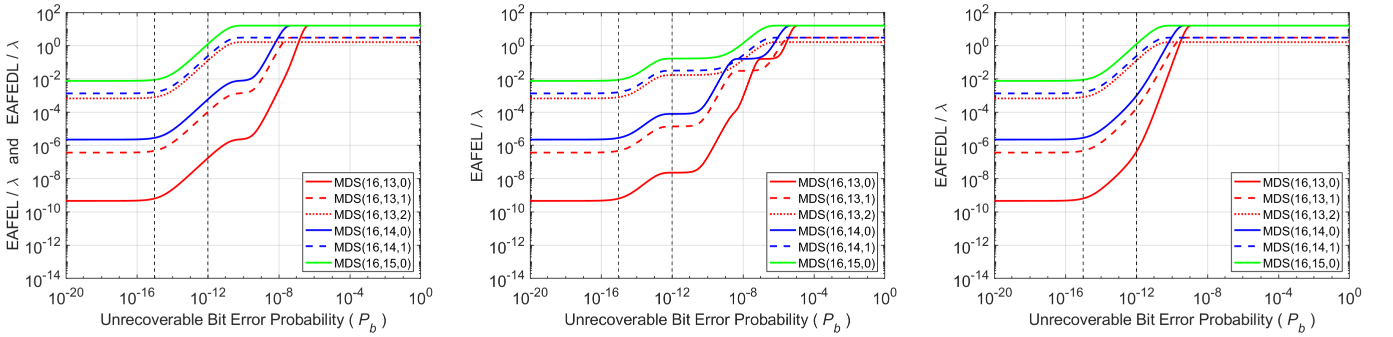
Clearly, the vulnerability of entities to loss increases with their size, which implies that lost entities are most likely large rather than small. For the case of the bimodal entity sizes, and for $v_2 \approxeq 0.01$, the number of the large 1-TB entities is significantly smaller than that of the 1-MB 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 8(a), 8(b) 8(c) that show the reliabil-



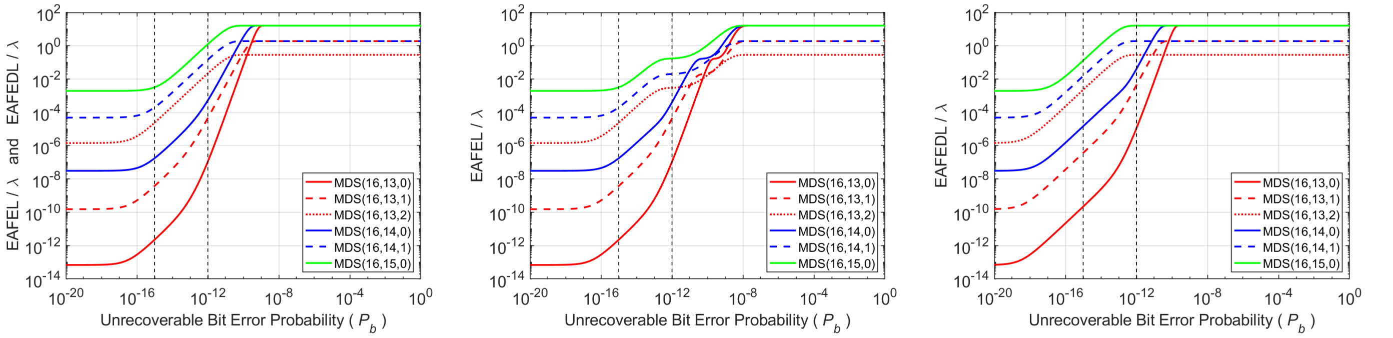
(a) Fixed Entity Size: $e_s = 10$ GB (b) Bimodal Entity Sizes: $e_{s,1} = 1$ MB, $e_{s,2} = 1$ TB (c) Bimodal Entity Sizes: $e_{s,1} = 1$ MB, $e_{s,2} = 1$ TB

Figure 7. 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 8. Normalized EAFEL and EAFEDL vs. P_b for various MDS(m, l, d) codes; symbol size $s = 512$ B, clustered 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 9. Normalized EAFEL and EAFEDL vs. P_b for various MDS(m, l, d) codes; symbol size $s = 5$ MB, declustered data placement.

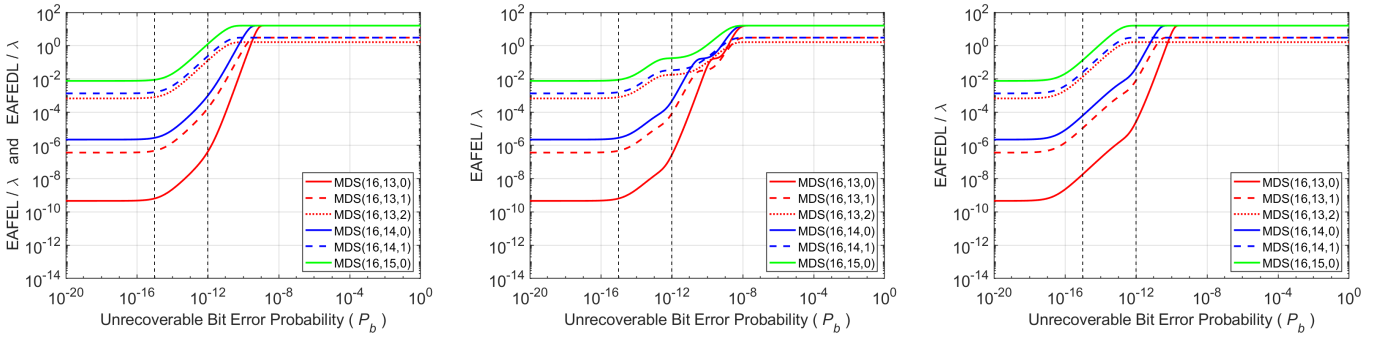
ity results for the case of clustered placement, we arrive to the same conclusions. From the above discussion, it follows that in the case of variable size entities, it is important to consider the EAFEDL rather than the EAFEL metric.

The effect of symbol size on reliability is assessed by considering the case of a large 5-MB symbol size. The corresponding normalized EAFEL/λ and EAFEDL/λ reliability metrics are shown in Figures 9 and 10. As expected, comparing these results with those shown in Figures 7 and 8, 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 [9] that was considered in [10] and listed in Table III. For the file sizes uniformly distributed within the bins, the mean is equal to 843 MB, the standard deviation to 2.8 GB and the second moment to 8.9 GB^2. 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 $L = 38$. In this case, the mean is equal to 843 MB, the standard deviation to 2.8 GB and the second moment to 8.5 GB^2. The corresponding reliability results are shown in Figures 11 and 12. In all cases considered, the reliability level achieved by the declustered data placement scheme is higher than that of the clustered one.

VI. 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



(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 $s = 5$ MB, clustered data placement.

TABLE III. CERN FILE SIZE DISTRIBUTION

	Bins			**Bin Mean Size**		**pdf**
j				$e_{s,j}$		v_j
1	1 B	–	2 B	2	B	0.00004559
2	2 B	–	5 B	4	B	0.00001275
3	5 B	–	10 B	8	B	0.00005533
4	10 B	–	22 B	16.0	B	0.00060401
5	22 B	–	46 B	34.0	B	0.00018569
6	46 B	–	100 B	73.0	B	0.00121244
7	100 B	–	215 B	157.5	B	0.00093013
8	215 B	–	464 B	339.5	B	0.00174431
9	464 B	–	1 KB	732.0	B	0.00675513
10	1 KB	–	2.154 KB	1.577	KB	0.00530524
11	2.154 KB	–	4.642 KB	3.398	KB	0.00496005
12	4.642 KB	–	10 KB	7.321	KB	0.00800625
13	10 KB	–	21.544 KB	15.772	KB	0.01174913
14	21.544 KB	–	46.416 KB	33.980	KB	0.01738480
15	46.416 KB	–	100 KB	73.208	KB	0.01359001
16	100 KB	–	215.443 KB	157.721	KB	0.01471745
17	215.443 KB	–	464.159 KB	339.801	KB	0.02018806
18	464.159 KB	–	1 MB	732.079	KB	0.02566358
19	1 MB	–	2.154 MB	1.577	MB	0.06221012
20	2.154 MB	–	4.642 MB	3.398	MB	0.07519022
21	4.642 MB	–	10 MB	7.321	MB	0.07654035
22	10 MB	–	21.544 MB	15.772	MB	0.09501620
23	21.544 MB	–	46.416 MB	33.980	MB	0.07847651
24	46.416 MB	–	100 MB	73.208	MB	0.07416942
25	100 MB	–	215.443 MB	157.721	MB	0.09371673
26	215.443 MB	–	464.159 MB	339.801	MB	0.08093624
27	464.159 MB	–	1 GB	732.079	MB	0.05399279
28	1 GB	–	2.154 GB	1.577	GB	0.04992384
29	2.154 GB	–	4.642 GB	3.398	GB	0.08871583
30	4.642 GB	–	10 GB	7.321	GB	0.03182476
31	10 GB	–	21.544 GB	15.772	GB	0.00452804
32	21.544 GB	–	46.416 GB	33.980	GB	0.00146156
33	46.416 GB	–	100 GB	73.208	GB	0.00017060
34	100 GB	–	215.443 GB	157.721	GB	0.00001375
35	215.443 GB	–	464.159 GB	339.801	GB	0.00000206
36	464.159 GB	–	1 TB	732.079	GB	0.00000069
37	1 TB	–	2.154 TB	1.577	TB	0.00000033
38	2.154 TB	–	4.310 TB	3.230	TB	0.00000001

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 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 EAFEL and the 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. It was demonstrated that an increased variability of entity sizes results in improved EAFEL, but degraded EAFEDL. We established that EAFEL and EAFEDL are adversely affected by the symbol size. 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.

APPENDIX

Proof of Propositions 1 and 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 (12) corresponds to $E(J)$ codewords, with $E(J)$ determined by (15). Consequently, for the number E_u of entities to be recovered it holds that

$$E_u \approx \frac{C_u}{E(J)} \,, \quad \text{for } u = d+1, \ldots, \tilde{r}-1 \,. \tag{42}$$

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 Equation (98) of [4], 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 (35), and f_{cor} accounts for the correlation of latent errors and is determined by Equation (29) of [4]. Consequently, the probability $\tilde{q}_u|K$ of loss of an entity that spans K codewords is determined by

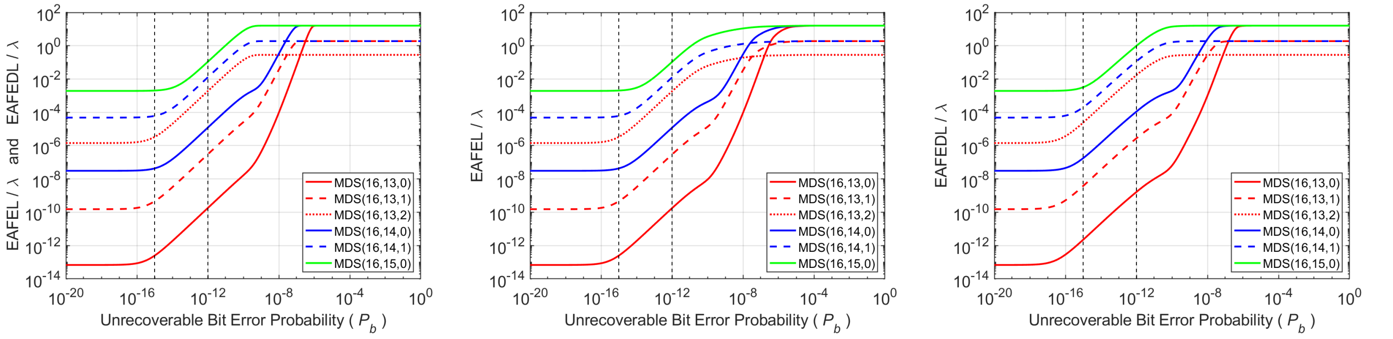
$$\tilde{q}_u|K \;=\; 1 - q_u^{f_{\text{cor}} K} \,. \tag{43}$$

Unconditioning (43) on K using (22) 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 (34). Thus, using (4), the probability $\tilde{q}_u(e_s)$ that the entity is lost is determined by

$$\tilde{q}_u(e_s) \;=\; \tilde{q}_{s,u}\left(\frac{e_s}{l\ s}\right) \,. \tag{44}$$

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

$$\check{q}_u(e_s) \;=\; e_s\, \tilde{q}_u(e_s) \,. \tag{45}$$

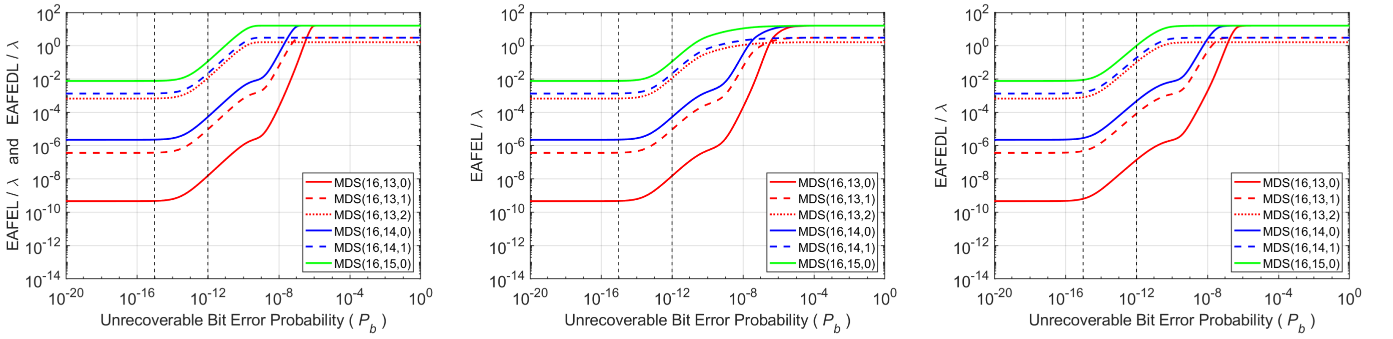


(a) Fixed File Size: $e_s = 843$ MB

(b) CERN File Sizes; $E(e_s) = 843$ MB

(c) CERN File Sizes; $E(e_s) = 843$ MB

Figure 11. Normalized EAFEL and EAFEDL vs. $P_{\rm b}$ 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 12. Normalized EAFEL and EAFEDL vs. $P_{\rm b}$ for various MDS(m, l, d) codes; symbol size $s = 512$ B, clustered data placement.

From (11), the probability $\tilde{q}_u$ that an arbitrary entity is lost is

$$\tilde{q}_u = \sum_{j=1}^{L} \tilde{q}_u(e_{s,j})\, v_j \ , \tag{46}$$

which, using (44), yields (33).

Similarly, from (11), it follows that the expected amount $\breve{q}_u$ of lost user data of an arbitrary entity is determined by

$$\breve{q}_u = \sum_{j=1}^{L} \breve{q}_u(e_{s,j})\, v_j \ , \tag{47}$$

which, using (44) and (45), yields (40).

Let $Y_{\rm U}$ be the number of lost entities and $\breve{Q}_{\rm U}$ the amount of lost user data at exposure level u during the rebuild process of the C_u codewords. Then it holds that,

$$E(Y_{\rm U}|C_u) \ = \ E_u\ \tilde{q}_u \ \overset{(42)}{\approx} \ \frac{C_u}{E(J)}\ \tilde{q}_u \ , \tag{48}$$

and
$$E(\breve{Q}_{\rm U}|C_u) \ = \ E_u\ \breve{q}_u \ \overset{(42)}{\approx} \ \frac{C_u}{E(J)}\ \breve{q}_u \ . \tag{49}$$

Note that $E(Y_{\rm U}|C_u)$, as determined by (48), can be obtained from Equation (71) of [6] by replacing the shard size J with its average value $E(J)$. Consequently, (32) and (36) are obtained from the corresponding Equations (42) and (44) of [6] by replacing the shard size J with its average value $E(J)$.

Note also that $E(\breve{Q}_{\rm U}|C_u)$, as determined by (49), can be obtained from (48) 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, (39) is obtained from (32) by replacing $\tilde{q}_u$ with $\breve{q}_u$.

References

[1] 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.

[2] 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.

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

[4] ——, "Reliability evaluation of erasure-coded storage systems with latent errors," ACM Trans. Storage, vol. 19, no. 1, Jan. 2023, pp. 1–47.

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

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

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

[8] 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: March, 2024]

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

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