Reliability Evaluation of Erasure Coded Systems

Ilias Iliadis and Vinodh Venkatesan
IBM Research – Zurich
8803 Rüschlikon, Switzerland
Email: ili@zurich.ibm.com, vinodh.iitm@gmail.com

***Abstract*—Replication is widely used to enhance the reliability of storage systems and protect data from device failures. The effectiveness of the replication scheme has been evaluated based on the Mean Time to Data Loss** (MTTDL) **and the Expected Annual Fraction of Data Loss** (EAFDL) **metrics. To provide high data reliability at high storage efficiency, modern systems employ advanced erasure coding redundancy and recovering schemes. This article presents a general methodology for obtaining the** EAFDL **and** MTTDL **of erasure coded systems analytically for arbitrary rebuild time distributions and for the symmetric, clustered, and declustered data placement schemes. Our analysis establishes that the declustered placement scheme offers superior reliability in terms of both metrics. The analytical results obtained enable the derivation of the optimal codeword lengths that maximize the** MTTDL **and minimize the** EAFDL**. It is theoretically shown that, for large storage systems that use a declustered placement scheme, both metrics are optimized when the codeword length is about 60% of the storage system size.**

Keywords–Reliability metric; MTTDL; EAFDL; RAID; MDS codes; Information Dispersal Algorithm; Prioritized rebuild.

I. INTRODUCTION

The reliability of storage systems is affected by data losses due to device and component failures, including disk and node failures. Permanent loss of data is prevented by deploying redundancy schemes that enable data recovery. However, additional device failures that may occur during rebuild operations could lead to permanent data losses. Over the years, several redundancy and recovery schemes have been developed to enhance the reliability of storage systems. These schemes offer different levels of reliability, with varying corresponding overheads due to the additional operations that need to be performed, and different levels of storage efficiencies that depend on the additional amount of redundant (parity) data that needs to be stored in the system [1].

The effectiveness of the redundancy schemes has been evaluated predominately based on the Mean Time to Data Loss (MTTDL) metric. Closed-form reliability expressions are typically obtained using Markov models, with the underlying assumption that the times to component failures and the rebuild times are independent and exponentially distributed [2-14]. Recent work has shown that these results also hold in the practical case of non-exponential failure time distributions. This was achieved based on a methodology for obtaining MTTDL that does not involve any Markov analysis [15]. The MTTDL metric has been used extensively to assess tradeoffs, to compare schemes, and to estimate the effect of various parameters on system reliability [16-20].

To cope with data losses encountered in the case of distributed and cloud storage systems, data is replicated and recovery mechanisms are used. For instance, Amazon S3 is designed to provide 99.999999999% (eleven nines) durability of data over a given year [21]. Similarly, also Facebook [22], LinkedIn [23] and Yahoo! [24] consider the amount of data lost in given periods. To address this issue, a recent work has introduced the Expected Annual Fraction of Data Loss (EAFDL) metric [25]. It has also presented a methodology for deriving this metric analytically in the case of replication-based storage systems, where user data is replicated r times and the copies are stored in different devices. As an alternative to replication, storage systems use advanced erasure codes that provide a high data reliability as well as a high storage efficiency. The use of such erasure codes can be traced back to as early as the 1980s when they were applied in systems with redundant arrays of inexpensive disks (RAID) [2][3]. The RAID-5, RAID-6 and replication-based systems are special cases of erasure coded systems. State-of-the-art data storage systems [26][27] employ more general erasure codes, where the choice of the codes used greatly affects the performance, reliability, and the storage and reconstruction overhead of the system. In this article, we focus on the reliability assessment of erasure coded systems and how the choice of codes affects the reliability in terms of the MTTDL and EAFDL metrics.

The MTTDL of erasure coded systems has been obtained analytically in [28]. It was theoretically shown that the MTTDL of erasure coded systems is practically insensitive to the distribution of the device failure times, but sensitive to the distribution of the device rebuild times. Simulation results confirmed the validity of the theoretical model. In this article, we establish that this also holds for the EAFDL metric. To reduce the amount of data lost, it is imperative to assess not only the frequency of data loss events, which is obtained through the MTTDL metric, but also the amount of data lost, which is expressed by the EAFDL metric [25]. The EAFDL and MTTDL metrics provide a useful profile of the size and frequency of data losses. Accordingly, we present a general framework and methodology for deriving the EAFDL analytically, along with the MTTDL, for erasure coded storage systems. The model developed captures the effect of the various system parameters as well as the effect of various codeword placement schemes, such as clustered, declustered, and symmetric data placement schemes. The results obtained show that the declustered placement scheme offers superior reliability in terms of both metrics. We also investigate the effect of the codeword length and identify the optimal values that offer the best reliability.

The key contributions of this article are the following. We consider the reliability of erasure coded systems that was assessed in our earlier work [1] for deterministic rebuild times. In this study, we extend our previous work by also considering

arbitrary rebuild times. We show that the codeword lengths that optimize the MTTDL and EAFDL metrics are similar. Subsequently, we derive the asymptotic analytic expressions for the MTTDL and EAFDL reliability metrics when the number of devices becomes large. We then obtain analytically the optimal codeword lengths corresponding to large storage systems. We establish theoretically that, for large storage systems that use a declustered placement scheme, both metrics are optimized when the codeword length is about 60% of the storage system size.

The remainder of the paper is organized as follows. Section II provides a survey of the relevant literature on erasure coded systems. Section III describes the storage system model and the corresponding parameters considered. Section IV presents the general framework and methodology for deriving the MTTDL and EAFDL metrics analytically for the case of erasure coded systems. Closed-form expressions for the symmetric, clustered, and declustered placement schemes are derived. Section V compares these schemes and establishes that the declustered placement scheme offers superior reliability. Section VI presents a thorough comparison of the reliability achieved by the declustered placement scheme under various codeword configurations. Finally, we conclude in Section VII.

II. Related Work

A comparison between erasure coding and replication in terms of availability in peer-to-peer systems was presented in [29] and [30]. These works established that erasure codes use an order of magnitude less storage than replication for systems with a similar level of reliability. Erasure codes, however, are more demanding as they may require Galois field arithmetic for encoding and decoding. Therefore, to improve the performance of erasure coded systems, new codes as well as new encoding and decoding techniques have been developed (see [31] and references therein).

The study performed in [30] was conducted by considering a dynamic environment where nodes join and leave the system and subsequently trigger data movement. In this context, it was argued that bandwidth, and not spare storage, is most likely the limiting factor for the scalability of peer-to-peer storage systems. Furthermore, not only do erasure codes introduce a higher complexity in the system owing to the encoding and decoding process, but also the entire task of maintaining redundancy in such a dynamic environment becomes more challenging. In contrast to these works that consider the codeword lengths being equal to the number of nodes, our work relaxes this constraint by considering codeword lengths that may be smaller than the number of nodes. This is desirable for performance reasons given that in real storage systems the lengths of the erasure codes used are kept constant and small, whereas the number of nodes grows with the system capacity. In addition, having a smaller code length then allows the use of different placement schemes, some of which enable faster rebuilds and hence a higher reliability for the same erasure code.

In [15],[25],[28],[32] and [33], it was shown that the replica and codeword placements can have a significant impact on reliability. For this reason we also consider and assess the effect of several codeword placement schemes in this article.

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 ($m > l$)
s	symbol size
(l, m)	MDS-code structure
k	spread factor of the data placement scheme
b	average reserved rebuild bandwidth per device
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
Y_i	lifetime of the ith device ($i = 1, \ldots, n$)
$F_Y(.)$	cumulative distribution function of Y_i ($i = 1, \ldots, n$)
s_{eff}	storage efficiency of redundancy scheme ($s_{\text{eff}} = l/m$)
U	amount of user data stored in the system ($U = s_{\text{eff}}\, n\, c$)
$\tilde{r}$	minimum number of codeword symbols lost that lead to an irrecoverable data loss ($\tilde{r} = m - l + 1$ and $2 \leq \tilde{r} \leq m$)
$f_X(.)$	probability density function of X ($f_X(.) = F'_X(.)$)
$1/\mu$	mean time to read (or write) an amount c of data at a rate b from (or to) a device ($1/\mu = E(X) = c/b$)
$1/\lambda$	mean time to failure of a storage device ($1/\lambda = E(Y_i)$)

III. Storage System Model

The storage system considered comprises n storage devices (nodes or disks), with each device storing an amount c of data, such that the total storage capacity of the system is $n\,c$. Modern data storage systems use various forms of data redundancy to protect data from device failures. When devices fail, the redundancy of the data affected is reduced and eventually lost. To avoid irrecoverable data loss, the system performs rebuild operations that use the data stored in the surviving devices to reconstruct the temporarily lost data, thus maintaining the initial data redundancy.

A. Redundancy

According to the erasure coded schemes considered, the user data is divided into blocks (or symbols) of a fixed size (e.g., sector size of 512 bytes) and complemented with parity symbols to form codewords. In this article, we consider (l, m) maximum distance separable (MDS) erasure codes, which are a mapping from l user data symbols to a set of m ($> l$) symbols, called a codeword, in such a way that any subset containing l of the m symbols of the codeword can be used to decode (reconstruct, recover) the codeword. The corresponding storage efficiency, s_{eff}, is given by

$$s_{\text{eff}} = \frac{l}{m}\ . \tag{1}$$

Consequently, the amount of user data, U, stored in the system is given by

$$U = s_{\text{eff}}\, n\, c\ =\ \frac{l\, n\, c}{m}\ . \tag{2}$$

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

The m symbols of each codeword are stored on m distinct devices, such that the system can tolerate any $\tilde{r} - 1$ device failures, but $\tilde{r}$ device failures may lead to data loss, with

$$\tilde{r} = m - l + 1\ . \tag{3}$$

From the preceding, it follows that

$$1 \leq l < m \qquad \text{and} \qquad 2 \leq \tilde{r} \leq m\ . \tag{4}$$

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 $s_{\text{eff}}^{(\text{replication})}=1/r$.
RAID-5: A RAID-5 array comprised of N devices uses an $(N-1,N)$-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 $(N-2,N)$-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 $s_{\text{eff}}^{(\text{RAID-6})}=(N-2)/N$.
Reed–Solomon: It is based on (l,m)-MDS erasure codes.

B. Symmetric Codeword Placement

We consider a placement where each codeword is stored on m distinct devices with one symbol per device. In a large storage system, the number of devices, n, is typically much larger than the codeword length, m. Therefore, there exist many ways in which a codeword of m symbols can be stored across a subset of the n devices. For each device in the system, let its *redundancy spread factor* k denote the number of devices over which the codewords stored on that device are spread [28]. The system effectively comprises n/k disjoint groups of k devices. Each group contains an amount U/k of user data, with the corresponding codewords placed on the corresponding k devices in a distributed manner. Each codeword is placed entirely in one of the n/k groups. Within each group, all $\binom{k}{m}$ possible ways of placing m symbols across k devices are equally used to store all the codewords in that group.

In such a symmetric placement scheme, within each of the n/k groups, the $m-1$ codeword symbols corresponding to the data on each device are *equally* spread across the remaining $k-1$ devices, the $m-2$ codeword symbols corresponding to the codewords shared by any two devices are equally spread across the remaining $k-2$ devices, and so on. Note also that the n/k groups are logical and therefore need not be physically located in the same node/rack/datacenter.

We proceed by considering the clustered and declustered placement schemes, which are special cases of symmetric placement schemes for which k is equal to m and n, respectively. This results in n/m groups for clustered and one group for declustered placement schemes.

1) Clustered Placement: In this placement scheme, the n devices are divided into disjoint sets of m devices, referred to as *clusters*. According to the *clustered* placement, each codeword is stored across the devices of a particular cluster, as shown in Figure 1. In such a placement scheme, it can be seen that no cluster stores the redundancies that correspond to data stored on another cluster. The entire storage system can essentially be modeled as consisting of n/m independent clusters. In each cluster, data loss occurs when $\tilde{r}$ devices fail successively before rebuild operations complete successfully.

2) Declustered Placement: In this placement scheme, all $\binom{n}{m}$ possible ways of placing m symbols across n devices are equally used to store all the codewords in the system, as shown in Figure 1.

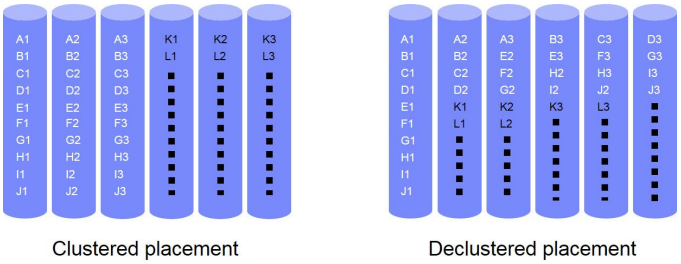


Figure 1. Clustered and declustered placement of codewords of length $m=3$ on $n=6$ devices. X1, X2, X3 represents a codeword (X = A, B, C, ..., L).

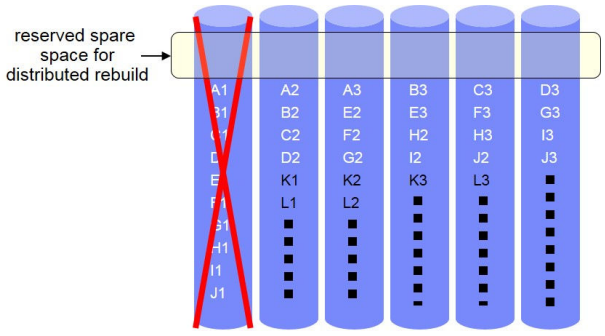
These two placement schemes represent the two extremes in which the symbols of the codewords associated with the data stored on a failing device are spread across the remaining devices and hence the extremes of the degree of parallelism that can be exploited when rebuilding this data. For declustered placement, the symbols are spread equally across *all* remaining devices, whereas for clustered placement, the symbols are spread across the smallest possible number of devices.

C. Codeword Reconstruction

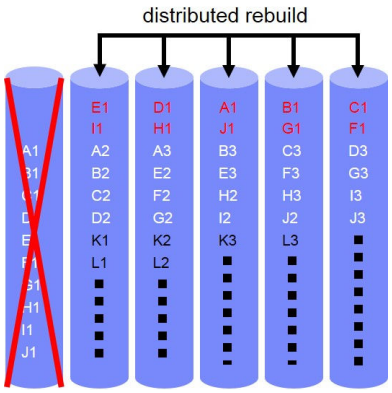
When storage devices fail, codewords lose some of their symbols, and this leads to a reduction in data redundancy. The system attempts to maintain its redundancy by reconstructing the lost codeword symbols using the surviving symbols of the affected codewords.

When a declustered placement scheme is used, as shown in Figure 2, spare space is reserved on each device for temporarily storing the reconstructed codeword symbols before they are transferred to a new replacement device. The rebuild bandwidth available on all surviving devices is used to rebuild the lost symbols in parallel. During this process, it is desirable to reconstruct the lost codeword symbols on devices in which another symbol of the same codeword is not already present. A similar reconstruction process is used for other symmetric placement schemes within each group of k devices, except for the clustered placement. When clustered placement is used, the codeword symbols are spread across all $k=m$ devices in each group (cluster). Therefore, reconstructing the lost symbols on the surviving devices of a group will result in more than one symbol of the same codeword on the same device. To avoid this, the lost symbols are reconstructed directly in spare devices as shown in Figure 3. In these reconstruction processes, decoding and re-encoding of data are assumed to be done on the fly and so the time taken for reconstruction is equal to the time taken to read and write the required data to the devices. Alternative methods of reconstruction based on regenerating codes have been proposed as a solution to reduce the amount of data transferred over the storage network during reconstruction (see [34] and references therein). They can, however, result in higher amounts of data being read from the surviving devices and therefore in longer rebuild times. The effect of these methods on the system reliability is outside the scope of this paper and is a subject of further investigation.

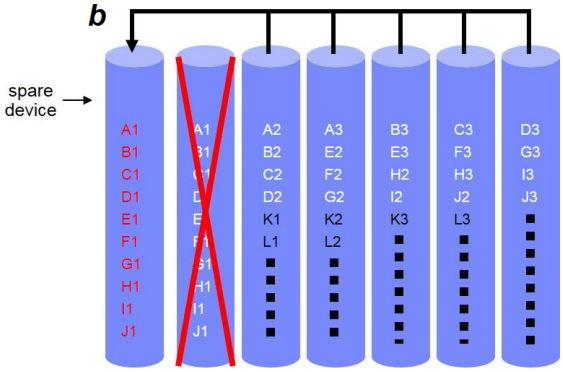
1) Exposure Levels and Amount of Data to Rebuild: At time t, let $D_j(t)$ be the number of codewords that have lost j



(a) Spare space reserved in each device.



(b) Distributed rebuild.



(c) Restoration of data on a spare device.

Figure 2. Rebuild under declustered placement.

symbols, with $0 \leq j \leq \tilde{r}$. The system is at exposure level u $(0 \leq u \leq \tilde{r})$, where

$$u = \max_{D_j(t)>0} j. \tag{5}$$

In other words, the system is at exposure level u if there are codewords with $m-u$ symbols left, but there are no codewords with fewer than $m-u$ symbols left in the system, that is, $D_u(t) > 0$, and $D_j(t) = 0$, for all $j > u$. These codewords are referred to as the *most-exposed* codewords. At $t = 0$, $D_j(0) = 0$, for all $j > 0$, and $D_0(0)$ is the total number of codewords stored in the system. Device failures and rebuild processes cause the values of $D_1(t), \cdots, D_{\tilde{r}}(t)$ to change over time, and when a data loss occurs, $D_{\tilde{r}}(t) > 0$. Device failures cause transitions to higher exposure levels, whereas rebuilds cause transitions to lower ones. Let t_u denote the time of the first transition from exposure level $u-1$ to exposure level u, and

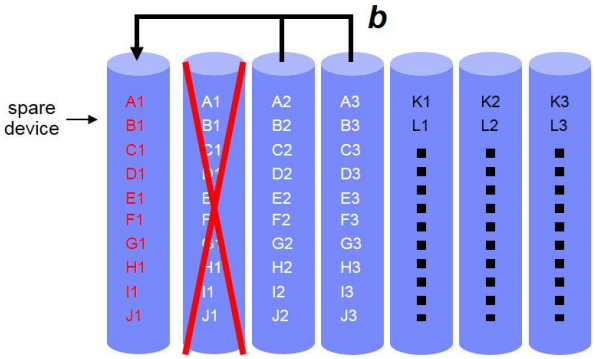


Figure 3. Rebuild under clustered placement.

TABLE II. NOTATION OF SYSTEM PARAMETERS AT EXPOSURE LEVELS

Parameter	Definition
u	exposure level
C_u	number of most-exposed codewords when entering exposure level u
R_u	rebuild time at exposure level u
$P_{u\rightarrow u+1}$	transition probability from exposure level u to $u+1$
$\tilde{n}_u$	number of devices at exposure level u whose failure causes an exposure level transition to level $u+1$
α_u	fraction of the rebuild time R_u still left when another device fails causing the exposure level transition $u \rightarrow u+1$
V_u	fraction of the most-exposed codewords that have symbols stored on another of the $\tilde{n}_u$ devices
A_u	amount of data corresponding to the C_u symbols ($A_u = C_u\, s$)
b_u	average rate at which recovered data is written at exposure level u

t_u^+ the instant immediately after t_u. Then, the number of most exposed codewords when entering exposure level u, denoted by C_u, $u = 1, \ldots, \tilde{r}$, is given by $C_u = D_u(t_u^+)$.

In Section IV-A1, we will derive the reliability metrics of interest using the direct path approximation, which considers only transitions from lower to higher exposure levels [15][28][32]. This implies that each exposure level is entered only once.

2) Prioritized or Intelligent Rebuild: At each exposure level u, the *prioritized or intelligent* rebuild process attempts to bring the system back to exposure level $u-1$ by recovering one of the u symbols that each of the most-exposed codewords has lost, that is, by recovering a total number of C_u symbols. Let A_u denote the amount of data corresponding to the C_u symbols and let s denote the symbol size. Then, it holds that

$$A_u = C_u\, s\,. \tag{6}$$

The notation used is summarized in Table II. For an exposure level u $(< \tilde{r})$, A_u represents the amount of data that needs to be rebuilt at that exposure level. In particular, upon the first-device failure, it holds that

$$A_1 = c\,, \tag{7}$$

which, combined with (6), implies that

$$C_1 = A_1/s = c/s\,. \tag{8}$$

D. Rebuild Process

During the rebuild process, a certain proportion of the device bandwidth is reserved for data recovery, with b denoting the actual average reserved rebuild bandwidth per device. The

average rebuild bandwidth is usually only a fraction of the total bandwidth available at each device; the remainder is used to serve user requests. Let us denote by b_u ($\leq b$) the average rate at which the amount A_u of data that needs to be rebuilt at exposure level u is written to selected device(s). Also, denote the cumulative distribution function of the time X required to read (or write) an amount c of data from (or to) a device by $F_X(.)$ and its corresponding probability density function by $f_X(.)$. The kth moment of X, $E(X^k)$, is then given by

$$E(X^k) = \int_0^\infty t^k f_X(t)dt \ , \quad \text{for } k = 1, 2, \dots \ . \tag{9}$$

In particular, let us denote by $1/\mu$ the average time required to read (or write) an amount c of data from (or to) a device, given by

$$\frac{1}{\mu} \triangleq E(X) = \frac{c}{b} \ . \tag{10}$$

E. Failure and Rebuild Time Distributions

In this work, we assume that the lifetimes $Y_1, \cdots, Y_n$ of the n devices are independent and identically distributed, with a cumulative distribution function $F_Y(.)$ and a mean of $1/\lambda$. In practice, this assumption is valid when the symbols of a codeword are placed on independently failing devices, for example, on devices located on different nodes/racks/datacenters. An extension of the analysis to also address correlated failures is part of future work. We further consider storage devices with failure time distributions that belong to the large class defined in [15], which includes real-world distributions, such as Weibull and gamma, as well as exponential distributions. The storage devices are *highly reliable* when 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 small, that is, when

$$\frac{\lambda}{\mu} = \frac{\lambda\, c}{b} \ll 1 \ . \tag{11}$$

According to [15][28], when the cumulative distribution functions F_Y and F_X satisfy the condition

$$\mu \int_0^\infty F_Y(t)[1 - F_X(t)]dt \ll 1, \quad \text{with } \frac{\lambda}{\mu} \ll 1 \ , \tag{12}$$

the MTTDL reliability metric of replication-based or erasure coded storage systems tends to be insensitive to the device failure distribution, that is, the MTTDL depends only on its mean $1/\lambda$, but not on its density $F_Y(.)$. In [25], it was shown that this also holds for the EAFDL metric in the case of replication-based storage systems and when the rebuild times are deterministic. In this article, we will show that this also holds for the EAFDL metric in the case of erasure coded systems under variable rebuild times.

IV. Derivation of MTTDL and EAFDL

We briefly review the general methodology for deriving the MTTDL and EAFDL metrics presented in [25]. This methodology does not involve any Markov analysis and holds for general failure time distributions, which can be exponential or non-exponential, such as the Weibull and gamma distributions that satisfy condition (12).

At any point in time, the system can be thought to be in one of two modes: normal mode and rebuild mode. During normal mode, all data in the system has the original amount of redundancy and there is no active rebuild process. During rebuild mode, some data in the system has less than the original amount of redundancy and there is an active rebuild process that is trying to restore the lost redundancy. A transition from normal mode to rebuild mode occurs when a device fails; we refer to the device failure that causes this transition as a *first-device* failure. Following a first-device failure, a complex sequence of rebuild operations and subsequent device failures may occur, which eventually leads the system either to an irrecoverable data loss (DL) with probability P_{DL} or back to the original normal mode by restoring initial redundancy, which occurs with probability $1 - P_{\text{DL}}$.

Let T be a typical interval of a fully operational period, that is, the time interval from the time t that the system is brought to its original state until a subsequent first-device failure occurs. For a system comprising n devices with a mean time to failure of a device equal to $1/\lambda$, the expected duration of T is given by [25]

$$E(T) = 1/(n\,\lambda) \ , \tag{13}$$

and the MTTDL by

$$\text{MTTDL} \approx \frac{E(T)}{P_{\text{DL}}} = \frac{1}{n\,\lambda\,P_{\text{DL}}} \ . \tag{14}$$

Let H denote the corresponding amount of data lost conditioned on the fact that a data loss has occurred. The metric of interest, that is, the Expected Annual Fraction of Data Loss (EAFDL), is subsequently obtained as the ratio of the expected amount of data lost to the expected time to data loss normalized to the amount of user data [25]:

$$\text{EAFDL} = \frac{E(H)}{\text{MTTDL} \cdot U} \ , \tag{15}$$

with the MTTDL expressed in years. Let us also denote by Q the unconditional amount of data lost upon a first-device failure. Note that Q is unconditional on the event of a data loss occurring in that it is equal either to H if the system suffers a data loss prior to returning to normal operation or to zero otherwise, that is,

$$Q = \begin{cases} H \ , & \text{if DL} \\ 0 \ , & \text{if no DL} \ . \end{cases} \tag{16}$$

Therefore, the expected amount of data lost, $E(Q)$, upon a first-device failure is given by

$$E(Q) = P_{\text{DL}}\, E(H) \ . \tag{17}$$

From (14), (15) and (17), we obtain the EAFDL as follows:

$$\text{EAFDL} \approx \frac{E(Q)}{E(T) \cdot U} = \frac{n\,\lambda\,E(Q)}{U} \ , \tag{18}$$

with $E(T)$ and $1/\lambda$ expressed in years.

A. Reliability Analysis

From (14) and (18), it follows that the derivation of the MTTDL and EAFDL metrics requires the evaluation of P_{DL} and $E(Q)$, respectively. These quantities are derived by considering the direct path approximation [15][28][32], which, under

conditions (11) and (12), accurately assesses the reliability metrics of interest [13][14][15][25].

Next, we present the general outline of the methodology in more detail.

1) Direct Path to Data Loss: Consider the direct path of successive transitions from exposure level 1 to $\tilde{r}$. In [15][28][32], it was shown that P_{DL} can be approximated by the probability of the direct path to data loss, $P_{\text{DL,direct}}$, that is,

$$P_{\text{DL}} \approx P_{\text{DL,direct}} = \prod_{u=1}^{r-1} P_{u \to u+1}, \tag{19}$$

where $P_{u \to u+1}$ denotes the transition probability from exposure level u to $u+1$. The above approximation holds when storage devices are highly reliable, that is, it holds for arbitrary device failure and rebuild time distributions that satisfy conditions (11) and (12). In this case, the relative error tends to zero as λ/μ tends to zero [15].

As the direct path to data loss dominates the effect of all other possible paths to data loss considered together, it follows that the amount of data lost H can be approximated by that corresponding to the direct path:

$$H \approx H_{\text{direct}} \, . \tag{20}$$

Also, from (16) and (20) it follows that

$$Q \approx \begin{cases} H_{\text{direct}} \, , & \text{if DL follows the direct path} \\ 0 \, , & \text{otherwise} \, . \end{cases} \tag{21}$$

Consequently, to derive the amount of data lost, it suffices to proceed by considering the H and Q metrics corresponding to the direct path to data loss.

Note that the amount of data lost, H, is the amount of user data stored in the most-exposed codewords when entering exposure level $\tilde{r}$, which can no longer be recovered and therefore is irrecoverably lost. As the number of these codewords is equal to $C_{\tilde{r}}$ and each of these codewords contains l symbols of user data, it holds that

$$H = C_{\tilde{r}} \, l \, s \, , \tag{22}$$

and using (6),

$$H = l \, A_{\tilde{r}} \, . \tag{23}$$

2) Amount of Data to Rebuild and Rebuild Times at Each Exposure Level: We now proceed to derive the conditional values of the random variables of interest given that the system goes through the direct path to data loss. Let R_u denote the rebuild times of the most-exposed codewords at each exposure level in this path, and let α_u be the fraction of the rebuild time R_u still left when another device fails causing the exposure level transition $u \to u+1$. In [35, Lemma 2], it was shown that, for highly reliable devices satisfying conditions (11) and (12), α_u is approximately uniformly distributed between zero and one, that is,

$$\alpha_u \sim U(0,1), \quad u = 1, \ldots, \tilde{r} - 1 \, . \tag{24}$$

Let $\vec{\alpha}$ denote the vector $(\alpha_1, \ldots, \alpha_{\tilde{r}-1})$, $\vec{\alpha}_u$ the vector $(\alpha_1, \ldots, \alpha_u)$, $\vec{C}_u$ the vector $(C_1, \cdots, C_u)$ and $\vec{A}_u$ the vector $(A_1, \cdots, A_u)$. Clearly, for the rebuild schemes considered, the fraction α_u of the rebuild time R_u still left also represents the expected fraction of the most-exposed codewords not yet recovered upon the next device failure. Therefore, the expected number of most-exposed codewords that are not yet recovered is equal to $\alpha_u \, C_u$. Clearly, the fraction V_u of these codewords that have symbols stored on the newly failed device depends on the codeword placement scheme. Consequently, the expected number of the most-exposed codewords when entering exposure level $u+1$ is given by

$$E(C_{u+1} \,|\, \vec{\alpha}, \vec{C}_u) = V_u \, \alpha_u \, C_u, \; , \quad u = 1, \ldots, \tilde{r} - 1 \, , \tag{25}$$

with V_u depending only on the placement scheme. Similarly, from (6), it follows that the corresponding expected amount of data that is not yet rebuilt is equal to $\alpha_u \, A_u$. From (25), we deduce that

$$E(A_{u+1} \,|\, \vec{\alpha}, \vec{A}_u) = V_u \, \alpha_u \, A_u, \; , \quad u = 1, \ldots, \tilde{r} - 1 \, , \tag{26}$$

An expression for the expected amount of data to be rebuilt at each exposure level is given by the following proposition.

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

$$E(A_u \,|\, \vec{\alpha}_{u-1}) = c \prod_{j=1}^{u-1} V_j \, \alpha_j \, . \tag{27}$$

Proof: We will prove (27) by induction. For $u = 2$, (27) holds owing to (7) and (26). Suppose that (27) holds for $u = k$, that is,

$$E(A_k \,|\, \vec{\alpha}_{k-1}) = c \prod_{j=1}^{k-1} V_j \, \alpha_j \, . \tag{28}$$

We will show that (27) also holds for $u = k+1$, that is,

$$E(A_{k+1} \,|\, \vec{\alpha}_k) = c \prod_{j=1}^{k} V_j \, \alpha_j \, . \tag{29}$$

From (26) it holds that

$$E(A_{k+1} \,|\, \vec{\alpha}, \vec{A}_k) = E(A_{k+1} \,|\, \vec{\alpha}_k, A_k) = V_k \, \alpha_k \, A_k \, . \tag{30}$$

It also holds that

$$E(A_{k+1} \,|\, \vec{\alpha}_k) = E_{A_k \,|\, \vec{\alpha}_k}[E(A_{k+1} \,|\, \vec{\alpha}_k, \vec{A}_k)] \, . \tag{31}$$

Substituting (30) into (31) yields

$$E(A_{k+1} \,|\, \vec{\alpha}_k) = E_{A_k \,|\, \vec{\alpha}_k}(V_k \, \alpha_k \, A_k) = V_k \, \alpha_k \, E(A_k \,|\, \vec{\alpha}_k) \, . \tag{32}$$

Clearly, the number C_k of most-exposed codewords when entering exposure level k and the corresponding amount of data A_k does not depend on the fraction α_k of the rebuild time R_k still left when another device fails causing the exposure level transition $k \to k+1$. It therefore holds that $E(A_k \,|\, \vec{\alpha}_k) = E(A_k \,|\, \vec{\alpha}_{k-1})$, and (32) yields

$$E(A_{k+1} \,|\, \vec{\alpha}_k) = V_k \, \alpha_k \, E(A_k \,|\, \vec{\alpha}_{k-1}) \overset{(28)}{=} c \prod_{j=1}^{k} V_j \, \alpha_j \, . \tag{33}$$

■

Remark 1: From (27), it follows that the expected amount of data to be rebuilt at each exposure level do not depend on the duration of the rebuild times.

At exposure level 1, according to (7), the amount A_1 of data to be recovered is equal to c. Given that this data is recovered at an average rate of b_1 and that the time required to write an amount c of data at an an average rate of b is equal to X, it follows that the rebuild time R_1 is given by

$$R_1 = \frac{b}{b_1} X \, . \tag{34}$$

As the rebuild times are proportional to the amount of data to be rebuilt and are inversely proportional to the rebuild rates, it holds that

$$E\left(\frac{R_{u+1}}{R_u} \mid \vec{\alpha}, \vec{A}_u\right) = E\left(\frac{A_{u+1}}{A_u} \mid \vec{\alpha}, \vec{A}_u\right) \frac{b_u}{b_{u+1}} \, , \; u \geq 1 \, . \tag{35}$$

Using (26), (35) yields

$$E\left(\frac{R_{u+1}}{R_u} \mid \vec{\alpha}, \vec{A}_u\right) = V_u \, \alpha_u \, \frac{b_u}{b_{u+1}} \, , \quad u = 1, \ldots, \tilde{r} - 2 \, , \tag{36}$$

and conditioning on R_u,

$$E(R_{u+1} \,|\, \vec{\alpha}, \vec{A}_u, R_u) = V_u \, \alpha_u \, \frac{b_u}{b_{u+1}} \, R_u \, , \; u = 1, \ldots, \tilde{r} - 2 \, . \tag{37}$$

The above implies that of all the random variables involved in vectors $\vec{\alpha}$ and $\vec{A}_u$, only a_u and R_u are essential for determining $E(R_{u+1})$. We proceed by considering the mean $1/\mu_u$ of the rebuild time R_u conditioned on α_{u-1} and R_{u-1}:

$$1/\mu_u \triangleq E(R_u \,|\, R_{u-1}, \alpha_{u-1}), \quad u = 2, \ldots, \tilde{r} - 1 \, . \tag{38}$$

From (37) and (38), it follows that

$$1/\mu_u = G_{u-1} \, \alpha_{u-1} \, R_{u-1} \, , \quad \text{for } u = 2, \ldots, \tilde{r} - 1 \, , \tag{39}$$

where

$$G_u \triangleq \frac{b_u}{b_{u+1}} \, V_u \, , \quad u = 1, \ldots, \tilde{r} - 2 \, . \tag{40}$$

The distribution of R_u, given R_{u-1} and α_{u-1}, could be modeled in several ways. We proceed as in [28] by considering the model B presented in [15], according to which the rebuild time R_u is determined completely by R_{u-1} and α_{u-1} and no new randomness is introduced in the rebuild time at exposure level u, that is,

$$R_u \,|\, R_{u-1}, \alpha_{u-1} = 1/\mu_u \;\; w.p.\, 1, \text{ for } u = 2, \ldots, \tilde{r} - 1 \, , \tag{41}$$

which by virtue of (39) yields

$$R_u = G_{u-1} \, \alpha_{u-1} \, R_{u-1} \, , \quad \text{for } u = 2, \ldots, \tilde{r} - 1 \, . \tag{42}$$

Repeatedly applying (42) and using (40) yields

$$R_u = \frac{b_1}{b_u} \, R_1 \, \prod_{j=1}^{u-1} V_j \, \alpha_j \, , \quad u = 1, \ldots, \tilde{r} - 1 \, . \tag{43}$$

Let $\tilde{n}_u$ be the number of devices at exposure level u whose failure before the rebuild of the most-exposed codewords causes an exposure level transition to level $u+1$. Subsequently, the transition probability $P_{u \to u+1}$ from exposure level u to $u+1$ depends on the duration of the corresponding rebuild time R_u and the aggregate failure rate of these $\tilde{n}_u$ highly reliable devices, and is given by [15]

$$P_{u \to u+1} \approx \tilde{n}_u \, \lambda \, R_u \, , \quad \text{for } u = 1, \ldots, \tilde{r} - 1 \, . \tag{44}$$

Conditioning on R_1 and $\vec{\alpha}_{u-1}$, and substituting (43) into (44), yields

$$P_{u \to u+1}(R_1, \vec{\alpha}_{u-1}) \approx \tilde{n}_u \, \lambda \, \frac{b_1}{b_u} \, R_1 \, \prod_{j=1}^{u-1} V_j \, \alpha_j \, . \tag{45}$$

Approximate expressions for the probability of data loss, P_{DL}, and the expected amount of data lost, $E(Q)$, are subsequently obtained by the following propositions.

Proposition 2: It holds that

$$P_{\text{DL}} \approx (\lambda \, c)^{\tilde{r}-1} \, \frac{1}{(\tilde{r}-1)!} \, \frac{E(X^{\tilde{r}-1})}{[E(X)]^{\tilde{r}-1}} \, \prod_{u=1}^{\tilde{r}-1} \frac{\tilde{n}_u}{b_u} \, V_u^{\tilde{r}-1-u} \, , \tag{46}$$

where $E(X^{\tilde{r}-1})$ is obtained by (9).

Proof: See Appendix A. ■

Proposition 3: It holds that

$$E(Q) \approx \, l \, c \, (\lambda \, c)^{\tilde{r}-1} \, \frac{1}{\tilde{r}\,!} \, \frac{E(X^{\tilde{r}-1})}{[E(X)]^{\tilde{r}-1}} \, \prod_{u=1}^{\tilde{r}-1} \frac{\tilde{n}_u}{b_u} \, V_u^{\tilde{r}-u} \, , \tag{47}$$

where $E(X^{\tilde{r}-1})$ is obtained by (9).

Proof: See Appendix B. ■

3) Evaluation of $E(H)$: The expected amount $E(H)$ of data lost conditioned on the fact that a data loss has occurred is obtained from (17) as the ratio of $E(Q)$ to P_{DL}. Consequently, using (46) and (47), it follows that

$$E(H) = \frac{E(Q)}{P_{\text{DL}}} \approx \left(\frac{l}{\tilde{r}} \, \prod_{u=1}^{\tilde{r}-1} V_u\right) c \, . \tag{48}$$

Remark 2: From (48), it follows that the expected amount of data lost conditioned on the fact that a data loss has occurred does not depend on the duration of the rebuild times.

4) Evaluation of MTTDL *and* EAFDL*:* Substituting (46) into (14) yields

$$\text{MTTDL} \approx \frac{1}{n \, \lambda} \, \frac{(\tilde{r}-1)!}{(\lambda \, c)^{\tilde{r}-1}} \, \frac{[E(X)]^{\tilde{r}-1}}{E(X^{\tilde{r}-1})} \, \prod_{u=1}^{\tilde{r}-1} \frac{b_u}{\tilde{n}_u} \, \frac{1}{V_u^{\tilde{r}-1-u}} \, . \tag{49}$$

Substituting (2) and (47) into (18) yields

$$\text{EAFDL} \approx m \, \lambda \, (\lambda \, c)^{\tilde{r}-1} \, \frac{1}{\tilde{r}\,!} \, \frac{E(X^{\tilde{r}-1})}{[E(X)]^{\tilde{r}-1}} \, \prod_{u=1}^{\tilde{r}-1} \frac{\tilde{n}_u}{b_u} \, V_u^{\tilde{r}-u} \, . \tag{50}$$

B. Symmetric Placement

Here, we consider the case where the redundancy spread factor k is in the interval $m < k \leq n$. As discussed in Section III-C2, at each exposure level u, the *prioritized* rebuild process recovers one of the u symbols that each of the C_u most-exposed codewords has lost by reading $m - \tilde{r} + 1$ of the remaining symbols. Thus, there are C_u symbols to be recovered in total, which corresponds to an amount A_u of data. For the symmetric placement discussed in Section III-B, these symbols are recovered by reading $(m - \tilde{r} + 1) \, C_u$ symbols, which corresponds to an amount $(m - \tilde{r} + 1) \, A_u$

of data, from the $k-u$ surviving devices in the affected group. Note that these are precisely the devices at exposure level u whose failure before the rebuild of the most-exposed codewords causes an exposure level transition to level $u+1$. Consequently, it holds that

$$\tilde{n}_u^{\text{sym}} = k - u \, . \tag{51}$$

Furthermore, it is desirable to write the recovered symbols to the spare space of these devices in such a way that no symbol is written to a device in which another symbol corresponding to the same codeword is already present. This will ensure that whenever a device fails, no more than one symbol from any codeword is lost. Owing to the symmetry of the symmetric placement, the same amount of data is being read from each of the $\tilde{n}_u$ devices. Similarly, the same amount of data is being written to each of the $\tilde{n}_u$ devices. Consequently, the total average read/write rebuild bandwidth b of each device is split between the reads and the writes, such that the average read rate is equal to $(m-\tilde{r}+1)\,b/(m-\tilde{r}+2)$ and the average write rate is equal to $b/(m-\tilde{r}+2)$. Therefore, the total average write bandwidth, which is also the average rebuild rate b_u, is given by

$$b_u^{\text{sym}} = \frac{\tilde{n}_u}{m-\tilde{r}+2}\, b \, , \quad u = 1, \ldots, \tilde{r}-1 \, . \tag{52}$$

Once all lost codeword symbols have been recovered, they are transferred to a new replacement device.

When the system enters exposure level u, the number of most-exposed codewords that need to be recovered is equal to C_u, $u = 1, \ldots, \tilde{r}$. Upon the next device failure, the expected number of most-exposed codewords that are not yet recovered is equal to $\alpha_u\, C_u$. Owing to the nature of the symmetric codeword placement, the newly failed device stores codeword symbols corresponding to only a fraction

$$V_u^{\text{sym}} = \frac{m-u}{k-u} \, , \quad u = 1, \ldots, \tilde{r}-1 \, . \tag{53}$$

of these most-exposed, not yet recovered codewords.

Substituting (51), (52), and (53) into (49), (50), and (48), and using (3), yields

$$\text{MTTDL}_k^{\text{sym}} \approx \frac{1}{n\,\lambda} \left[\frac{b}{(l+1)\,\lambda\, c}\right]^{m-l} (m-l)! \, \frac{[E(X)]^{m-l}}{E(X^{m-l})} \prod_{u=1}^{m-l} \left(\frac{k-u}{m-u}\right)^{m-l-u} \, , \tag{54}$$

$$\text{EAFDL}_k^{\text{sym}} \approx \lambda \left[\frac{(l+1)\,\lambda\, c}{b}\right]^{m-l} \frac{m}{(m-l+1)!} \, \frac{E(X^{m-l})}{[E(X)]^{m-l}} \prod_{u=1}^{m-l} \left(\frac{m-u}{k-u}\right)^{m-l+1-u} \, , \tag{55}$$

and

$$E(H)_k^{\text{sym}} \approx \left(\frac{l}{m-l+1} \prod_{u=1}^{m-l} \frac{m-u}{k-u}\right) c \tag{56}$$

$$= \frac{l\,(m-1)!\,(k-m+l-1)!}{(m-l+1)\,(k-1)!\,(l-1)!}\, c \, . \tag{57}$$

Note that for a deterministic rebuild time distribution, for which it holds that $E(X^{m-l}) = [E(X)]^{m-l}$, and for a replication-based system, for which $m = r$ and $l = 1$, (54) and (55) are in agreement with Equations (42.b) and (43.b) of [25], respectively.

Remark 3: From (54), (55), and (56), it follows that $\text{MTTDL}_k^{\text{sym}}$ depends on n, but $\text{EAFDL}_k^{\text{sym}}$ and $E(H)_k^{\text{sym}}$ do not.

Remark 4: From (54), (55), and (56), it follows that, for $m-l=1$, $\text{MTTDL}_k^{\text{sym}}$ does not depend on k, whereas for $m-l>1$, $\text{MTTDL}_k^{\text{sym}}$ is increasing in k. Also, for $m-l \geq 1$, $\text{EAFDL}_k^{\text{sym}}$ and $E(H)_k^{\text{sym}}$ are decreasing in k. Consequently, within the class of symmetric placement schemes considered, that is, for $m < k \leq n$, the $\text{MTTDL}_k^{\text{sym}}$ is maximized and the $\text{EAFDL}_k^{\text{sym}}$ and the $E(H)_k^{\text{sym}}$ are minimized when $k = n$. Also, given that $E(X) = c/b$, the $\text{MTTDL}_k^{\text{sym}}$ and $\text{EAFDL}_k^{\text{sym}}$ depend on the $(m-l)$th moment of the rebuild time distribution, whereas $E(H)_k^{\text{sym}}$ does not depend on the rebuild times. Furthermore, given that $E(X^{m-l}) \geq [E(X)]^{m-l}$, random rebuild times result in lower MTTDL and higher EAFDL values than deterministic rebuild times do.

Approximate expressions for the reliability metrics of interest are given by the following propositions.

Proposition 4: For large values of k, m, l, and $m-l$, the $E(H)^{\text{sym}}$ normalized to c can be approximated as follows:

$$\log\left(E(H)_{\text{approx}}^{\text{sym}}/c\right) \approx \log\left(\frac{(1-h)\,x\,k}{h\,x\,k+1}\sqrt{\frac{1-h}{1-hx}}\right) + k\,V(h,x) \, , \tag{58}$$

where $V(h,x)$ is given by

$$V(h,x) \triangleq \log\left(\frac{x^x\,(1-hx)^{1-hx}}{[(1-h)x]^{(1-h)x}}\right) \, , \tag{59}$$

h is given by

$$h \triangleq 1 - s_{\text{eff}} = 1 - \frac{l}{m} \tag{60}$$

and x by

$$x \triangleq \frac{m}{k} \, . \tag{61}$$

Proof: See Appendix C. ■

Proposition 5: For large values of k, m, l, and $m-l$, the $\text{MTTDL}^{\text{sym}}$ normalized to $1/\lambda$ can be approximated as follows:

$$\log\left(\lambda\,\text{MTTDL}_{\text{approx}}^{\text{sym}}\right) \approx \log\left(\frac{k}{n}\right) + k^2\,\frac{W(h,x)}{2} + k\,hx\,\log\left(\frac{hx\sqrt{x}\,k\,b}{e\,[(1-h)x\,k+1]\,\lambda\,c}\right) - \frac{1}{8}\left[h(1-x) - \log\left(\frac{1-h}{1-hx}\right)\right] + \log\left(\sqrt{\frac{2\pi hx}{k}}\right) + \log\left(\frac{[E(X)]^{hxk}}{E(X^{hxk})}\right) \, , \tag{62}$$

where

$$W(h,x) \triangleq hx(1-x) - \log\left(\frac{\left[(1-h)^{(1-h)^2}\,x^{h^2}\right]^{x^2}}{(1-hx)^{(1-hx)^2}}\right) \, , \tag{63}$$

and h and x are given by (60) and (61), respectively.

Proof: See Appendix D. ■

Proposition 6: For large values of k, m, l, and $m-l$, the EAFDL$^{\text{sym}}$ normalized to λ can be approximated as follows:

$$\begin{aligned}\log\left(\text{EAFDL}^{\text{sym}}_{\text{approx}}/\lambda\right) \approx \\ -k^2\,\frac{W(h,x)}{2} + k\left\{hx\,\log\left(\frac{e\,[(1-h)x\,k+1]\,\lambda\,c}{h\,\sqrt{x}\,k\,b}\right)\right. \\ \left. + \log\left(\frac{(1-hx)^{1-hx}}{(1-h)^{(1-h)x}}\right)\right\} \\ + \frac{1}{8}\,h(1-x) + \log\left(\frac{1}{hx\,k+1}\,\sqrt{\frac{x\,k}{2\pi h}}\left(\frac{1-h}{1-hx}\right)^{\frac{3}{8}}\right) \\ + \log\left(\frac{E(X^{hxk})}{[E(X)]^{hxk}}\right)\,, \end{aligned} \tag{64}$$

where h, x, and $W(h,x)$ are given by (60), (61), and (63), respectively.

Proof: See Appendix E. ■

C. Clustered Placement

As discussed in Section III-B1, in the clustered placement scheme, the n devices are divided into disjoint sets of m devices, referred to as *clusters*. According to the *clustered* placement, each codeword is stored across the devices of a particular cluster. At each exposure level u, the rebuild process recovers one of the u symbols that each of the C_u most-exposed codewords has lost by reading $m-\tilde{r}+1$ of the remaining symbols. Note that the remaining symbols are stored on the $m-u$ surviving devices in the affected group. As these are precisely the devices at exposure level u whose failure before the rebuild of the most-exposed codewords causes an exposure level transition to level $u+1$, it holds that

$$\tilde{n}_u^{\text{clus}} = m-u\,. \tag{65}$$

The rebuild process in clustered placement recovers the lost symbols by reading $m-\tilde{r}+1$ symbols from $m-\tilde{r}+1$ of the $\tilde{n}_u$ surviving devices of the affected cluster. The lost symbols are computed on-the-fly and written to a spare device using the rebuild bandwidth at an average rate of b. Consequently, it holds that

$$b_u^{\text{clus}} = b\,, \quad u=1,\ldots,\tilde{r}-1\,. \tag{66}$$

Remark 5: Note that as far as the data placement is concerned, the clustered placement scheme is a special case of a symmetric placement scheme for which k is equal to m. However, its reliability assessment cannot be directly obtained from the reliability results derived in Section IV-B for the symmetric placement scheme by simply setting $k=m$. The reason for that is the difference in the rebuild processes. In the case of a symmetric placement scheme, recovered symbols are written to the spare space of existing devices, whereas in the case of a clustered placement scheme, recovered symbols are written to a spare device. This results in different rebuild bandwidths, which are given by (52) and (66), respectively.

When the system enters exposure level u, the number of most-exposed codewords that need to be recovered is equal to C_u, $u=1,\ldots,\tilde{r}$. Upon the next device failure, the expected number of most-exposed codewords that have not yet been recovered is equal to $\alpha_u\,C_u$. Clearly, all these codewords have symbols stored on the newly failed device, which implies that

$$V_u^{\text{clus}} = 1\,, \quad u=1,\ldots,\tilde{r}-1\,. \tag{67}$$

Substituting (65), (66), and (67) into (49), (50), and (48), and using (3), yields

$$\text{MTTDL}^{\text{clus}} \approx \frac{1}{n\,\lambda}\left(\frac{b}{\lambda\,c}\right)^{m-l}\frac{1}{\binom{m-1}{l-1}}\,\frac{[E(X)]^{m-l}}{E(X^{m-l})}\,, \tag{68}$$

$$\text{EAFDL}^{\text{clus}} \approx \lambda\left(\frac{\lambda\,c}{b}\right)^{m-l}\binom{m}{l-1}\,\frac{E(X^{m-l})}{[E(X)]^{m-l}}\,, \tag{69}$$

and

$$E(H)^{\text{clus}} = \frac{l}{m-l+1}\,c\,. \tag{70}$$

Note that the MTTDL derived in (68) is in agreement with Equation (15) of [28] (with $c/b=1/\mu$, $E(X)=M_1(G_\mu)$ and $E(X^{m-l})=M_{m-l}(G_\mu)$). For a RAID-5 array system, for which $n=m=N$ and $l=N-1$, and for a RAID-6 array system, for which $n=m=N$ and $l=N-2$, and for an exponential rebuild time distribution, for which it holds that $E(X^2)/[E(X)]^2=2$, Eq. (68) is in agreement with the MTTDL equations reported in [2][3]. Also, for a deterministic rebuild time distribution, for which it holds that $E(X^{m-l})=[E(X)]^{m-l}$, and for a replication-based system, for which $m=r$ and $l=1$, (68), (69), and (70) are in agreement with Equations (42.a), (43.a), and (39.a) of [25], respectively.

Remark 6: From (68), (69), and (70), and given that $E(X)=c/b$, the MTTDL$^{\text{clus}}$ and EAFDL$^{\text{clus}}$ depend on the $(m-l)$th moment of the rebuild time distribution, whereas $E(H)^{\text{clus}}$ does not depend on the rebuild times. Furthermore, given that $E(X^{m-l}) \geq [E(X)]^{m-l}$, random rebuild times result in lower MTTDL and higher EAFDL values than deterministic rebuild times do.

Approximate expressions for the reliability metrics of interest are given by the following propositions.

Proposition 7: For large values of n, m, l, and $m-l$, the MTTDL$^{\text{clus}}$ normalized to $1/\lambda$ and the EAFDL$^{\text{clus}}$ normalized to λ can be approximated as follows:

$$\lambda\,\text{MTTDL}^{\text{clus}}_{\text{approx}} \approx \sqrt{\frac{2\pi hx}{(1-h)n}}\left[\left(\frac{hb}{\lambda\,c}\right)^h(1-h)^{1-h}\right]^{xn} \frac{[E(X)]^{hxn}}{E(X^{hxn})}\,, \tag{71}$$

$$\text{EAFDL}^{\text{clus}}_{\text{approx}}/\lambda \approx \frac{1}{h}\sqrt{\frac{1-h}{2\pi hxn}}\left[\left(\frac{hb}{\lambda\,c}\right)^h(1-h)^{1-h}\right]^{-xn} \frac{E(X^{hxn})}{[E(X)]^{hxn}}\,, \tag{72}$$

where

$$x = \frac{m}{n}\,, \tag{73}$$

and h is given by (60).

Proof: From (68) and (69) it follows that

$$\text{MTTDL}^{\text{clus}} \approx \frac{1}{n\lambda}\left(\frac{b}{\lambda\, c}\right)^{m-l} \frac{(m-1)(m-l)!\, l!}{l\, m!} \frac{[E(X)]^{m-l}}{E(X^{m-l})}, \tag{74}$$

and

$$\text{EAFDL}^{\text{clus}} \approx \lambda\left(\frac{\lambda\, c}{b}\right)^{m-l} \frac{l\, m!}{(m-l+1)(m-l)!\, l!} \frac{E(X^{m-l})}{[E(X)]^{m-l}}. \tag{75}$$

Using Stirling's approximation for large values of n,

$$n! \approx \sqrt{2\pi n}\left(\frac{n}{e}\right)^n , \tag{76}$$

(74) and (75) yield

$$\text{MTTDL}^{\text{clus}} \approx \frac{1}{n\,\lambda}\left(\frac{b}{\lambda\, c}\right)^{m-l} \frac{m-1}{l}\sqrt{\frac{2\pi(m-l)\, l}{m}} \frac{(m-l)^{m-l}\, l^l}{m^m} \frac{[E(X)]^{m-l}}{E(X^{m-l})} , \tag{77}$$

and

$$\text{EAFDL}^{\text{clus}} \approx \lambda\left(\frac{\lambda\, c}{b}\right)^{m-l} \frac{l}{m-l+1}\sqrt{\frac{m}{2\pi(m-l)\, l}} \frac{m^m}{(m-l)^{m-l}\, l^l} \frac{E(X^{m-l})}{[E(X)]^{m-l}} . \tag{78}$$

From (1), (60), and (73), it follows that

$$l \;=\; s_{\text{eff}}\, m \;=(1-h)\, m \;=(1-h)\, x\, n \tag{79}$$

and

$$m-l \;=\; (1-s_{\text{eff}})\, m \;=\; h\, m \;=\; h\, x\, n\, . \tag{80}$$

Substituting (79) and (80) into (77) and (78) yields (71) and (72), respectively. ■

D. Declustered Placement

As discussed in Section III-B, the declustered placement scheme is a special case of a symmetric placement scheme in which k is equal to n. Consequently, for $k = n$, (54), (55), and (56) yield

$$\text{MTTDL}^{\text{declus}} \approx \frac{1}{n\,\lambda}\left[\frac{b}{(l+1)\,\lambda\, c}\right]^{m-l} (m-l)! \frac{[E(X)]^{m-l}}{E(X^{m-l})} \prod_{u=1}^{m-l}\left(\frac{n-u}{m-u}\right)^{m-l-u} , \tag{81}$$

$$\text{EAFDL}^{\text{declus}} \approx \lambda\left[\frac{(l+1)\,\lambda\, c}{b}\right]^{m-l} \frac{m}{(m-l+1)!} \frac{E(X^{m-l})}{[E(X)]^{m-l}} \prod_{u=1}^{m-l}\left(\frac{m-u}{n-u}\right)^{m-l+1-u} , \tag{82}$$

and

$$E(H)^{\text{declus}} \approx \left(\frac{l}{m-l+1}\prod_{u=1}^{m-l}\frac{m-u}{n-u}\right) c \tag{83}$$

$$= \frac{l\,(m-1)!\,(n-m+l-1)!}{(m-l+1)\,(n-1)!\,(l-1)!}\, c\, . \tag{84}$$

Note that the MTTDL derived in (81) is in agreement with Equation (16) of [28], with $c/b = 1/\mu$ and $[E(X)]^{m-l}/E(X^{m-l}) = M_1^{m-l}\left(G_{\frac{n-1}{l+1}\mu}\right)/M_{m-l}\left(G_{\frac{n-1}{l+1}\mu}\right)$. Also, for a deterministic rebuild time distribution, for which it holds that $E(X^{m-l}) = [E(X)]^{m-l}$, and for a replication-based system, for which $m = r$ and $l = 1$, (81), (82) and (83) are in agreement with Equations (36.b), (37.b), and (39.b) of [25], respectively.

Remark 7: From (81), (82), and (83), and given that $E(X) = c/b$, it follows that $\text{MTTDL}^{\text{declus}}$ and $\text{EAFDL}^{\text{declus}}$ depend on the $(m-l)$th moment of the rebuild time distribution, whereas $E(H)^{\text{clus}}$ does not depend on the rebuild times. Furthermore, given that $E(X^{m-l}) \geq [E(X)]^{m-l}$, random rebuild times result in lower MTTDL and higher EAFDL values than deterministic rebuild times do.

Approximate expressions for the reliability metrics of interest are given by the following propositions.

Proposition 8: For large values of n, m, l, and $m-l$, the $\text{MTTDL}^{\text{declus}}$ normalized to $1/\lambda$ can be approximated as follows:

$$\log\left(\lambda\,\text{MTTDL}^{\text{declus}}_{\text{approx}}\right) \approx n^2\,\frac{W(h,x)}{2} + n\,hx\,\log\left(\frac{hx\sqrt{x}\,n\,b}{e\,[(1-h)x\,n+1]\,\lambda\, c}\right) - \frac{1}{8}\left[h(1-x) - \log\left(\frac{1-h}{1-hx}\right)\right] + \log\left(\sqrt{\frac{2\pi hx}{n}}\right) + \log\left(\frac{[E(X)]^{hxn}}{E(X^{hxn})}\right) , \tag{85}$$

where h, x, and $W(h,x)$ are given by (60), (73), and (63), respectively.

Proof: Immediate from Proposition 5 by replacing k with n. ■

Proposition 9: For large values of n, m, l, and $m-l$, the $\text{EAFDL}^{\text{declus}}$ normalized to λ can be approximated as follows:

$$\log\left(\text{EAFDL}^{\text{declus}}_{\text{approx}}/\lambda\right) \approx -n^2\,\frac{W(h,x)}{2} + n\left\{hx\,\log\left(\frac{e\,[(1-h)x\,n+1]\,\lambda\, c}{h\,\sqrt{x}\,n\,b}\right) + \log\left(\frac{(1-hx)^{1-hx}}{(1-h)^{(1-h)x}}\right)\right\} + \frac{1}{8}\,h(1-x) + \log\left(\frac{1}{hx\,n+1}\sqrt{\frac{x\,n}{2\pi h}}\left(\frac{1-h}{1-hx}\right)^{\frac{3}{8}}\right) + \log\left(\frac{E(X^{hxn})}{[E(X)]^{hxn}}\right) , \tag{86}$$

where h, x, and $W(h,x)$ are given by (60), (73), and (63), respectively.

Proof: Immediate from Proposition 6 by replacing k with n and using (73). ■

Proposition 10: For large values of n, m, l, and $m-l$, the $E(H)^{\text{declus}}$ normalized to c can be approximated as follows:

$$\log\left(E(H)^{\text{declus}}_{\text{approx}}/c\right) \approx \log\left(\frac{(1-h)\,x\,n}{h\,x\,n+1}\sqrt{\frac{1-h}{1-hx}}\right) + n\,V(h,x)\ , \quad (87)$$

where h, x, and $V(h,x)$ are given by (60), (73), and (59), respectively.

Proof: Immediate from Proposition 4 by replacing k with n and using (73). ■

E. Accuracy of Approximations

Here, we assess the accuracy of the approximate reliability expressions derived by the preceding propositions. Regarding the MTTDL measure, we consider the ratio of the approximation $\text{MTTDL}^{\text{clus}}_{\text{approx}}$ given by (71) to $\text{MTTDL}^{\text{clus}}$ given by (68). Note that the ratio $\text{MTTDL}^{\text{clus}}_{\text{approx}}/\text{MTTDL}^{\text{clus}}$ only depends on m and l given that the approximation is obtained by only approximating the term $\frac{1}{\binom{m-1}{l-1}}$ that appears in (68). We also consider the ratio of the approximation $\text{EAFDL}^{\text{clus}}_{\text{approx}}$ given by (72) to $\text{EAFDL}^{\text{clus}}$ given by (69). Note that also this ratio only depends on m and l.

The ratios corresponding to the two reliability measures are shown in Figure 4 as a function of the codeword length for various storage efficiencies. As expected, for any given storage efficiency, for large values of m (and therefore l) the Stirling's approximation is accurate and therefore the ratio of the reliability measures approaches one. But even for small values of m, the ratios are close to one, which implies that the approximations are quite accurate.

Next, we consider the symmetric placement scheme. Regarding the MTTDL measure, we consider the ratio of the approximation $\text{MTTDL}^{\text{sym}}_{\text{approx}}$ given by (62) to $\text{MTTDL}^{\text{sym}}$ given by (81). Note that the ratio $\text{MTTDL}^{\text{sym}}_{\text{approx}}/\text{MTTDL}^{\text{sym}}$ only depends on k, m and l given that the approximation is obtained by only approximating the product $(m-l)!\prod_{u=1}^{m-l}\left(\frac{k-u}{m-u}\right)^{m-l-u}$ that appears in (81). We also consider the ratio of the approximation $\text{EAFDL}^{\text{sym}}_{\text{approx}}$ given by (64) to $\text{EAFDL}^{\text{sym}}$ given by (82) and the ratio of the approximation $E(H)^{\text{sym}}_{\text{approx}}$ given by (58) to $E(H)^{\text{sym}}$ given by (83). Note that also these ratios only depend on k, m and l. The ratios of the three measures are shown in Figures 5, 6 and 7 as a function of the codeword length for various spread factors and storage efficiencies. As expected, for any given storage efficiency, for large values of m (and therefore l) the Stirling's approximation of the $(m-l)!$ term is quite accurate. However, the approximations of the products $\prod_{u=1}^{m-l}\left(\frac{k-u}{m-u}\right)^{m-l-u}$ and $\prod_{u=1}^{m-l}\left(\frac{k-u}{n-u}\right)^{m-l+1-u}$ that appear in the MTTDL and EAFDL expressions in (81) and (82), respectively, result in ratios close to one only for large values of x. For small values of x, they yield ratios that tend to be insensitive as k increases. These ratios, however, still preserve the order of magnitude of the reliability measures. For example, in the case of $k = 200$, $s_{\text{eff}} = 1/2$ and $m = 4$ (which implies that $l = 2$), it holds that $\text{MTTDL}^{\text{sym}}_{\text{approx}}/\text{MTTDL}^{\text{sym}} = 0.913$, with $\text{MTTDL}^{\text{sym}}_{\text{approx}}$ being of the same order as $\text{MTTDL}^{\text{sym}}$ given that $n\,\lambda\,\text{MTTDL}^{\text{sym}}/k = 7.37\times10^{4}$ and $n\,\lambda\,\text{MTTDL}^{\text{sym}}_{\text{approx}}/k = 6.73\times10^{4}$. Also, in the case of $k = 200$, $s_{\text{eff}} = 1/5$ and $m = 5$ (which implies that $l = 1$), it holds that $\text{MTTDL}^{\text{sym}}_{\text{approx}}/\text{MTTDL}^{\text{sym}} = 0.884$, with $\text{MTTDL}^{\text{sym}}_{\text{approx}}$ being of the same order as $\text{MTTDL}^{\text{sym}}$ given that $n\,\lambda\,\text{MTTDL}^{\text{sym}}/k = 3.96\times10^{20}$ and $n\,\lambda\,\text{MTTDL}^{\text{sym}}_{\text{approx}}/k = 3.50\times10^{20}$. Furthermore, for the EAFDL metric it holds that in this case $\text{EAFDL}^{\text{sym}}_{\text{approx}}/\text{EAFDL}^{\text{sym}} = 1.206$, with $\text{EAFDL}^{\text{sym}}_{\text{approx}}$ being of the same order as $\text{EAFDL}^{\text{sym}}$ given that $\text{EAFDL}^{\text{sym}}/\lambda = 1.99\times10^{-31}$ and $\text{EAFDL}^{\text{sym}}_{\text{approx}}/\lambda = 2.40\times10^{-31}$. Consequently, the approximations are quite accurate.

V. Optimal Placement

Here, we identify which of the placement schemes considered offers the best reliability in terms of the MTTDL, EAFDL and $E(H)$ metrics. From Remark 4, it follows that the placement that maximizes MTTDL and minimizes EAFDL and $E(H)$ is either the clustered $(k = m)$ or the declustered one $(k = n)$. We therefore proceed by comparing these two schemes when $m \neq n$, that is, when $m < n$. This implies that we compare the two schemes when there are at least two clustered groups, that is, when $m \leq n/2$, or, by also using (3) and (4), when

$$1 \leq l < m \qquad \text{and} \qquad 1 \leq m-l < m \leq \frac{n}{2}\ . \quad (88)$$

A. *Maximizing* MTTDL

From (68) and (81), it follows that

$$\frac{\text{MTTDL}^{\text{declus}}}{\text{MTTDL}^{\text{clus}}} \approx \left(\frac{1}{l+1}\right)^{m-l}(m-l)!\binom{m-1}{l-1}\prod_{u=1}^{m-l}\left(\frac{n-u}{m-u}\right)^{m-l-u}\ . \quad (89)$$

Remark 8: From (89), it follows that the placement that maximizes MTTDL does not depend on λ, b and c nor on the rebuild time distribution.

Depending on the values of m and l, we consider the following three cases:

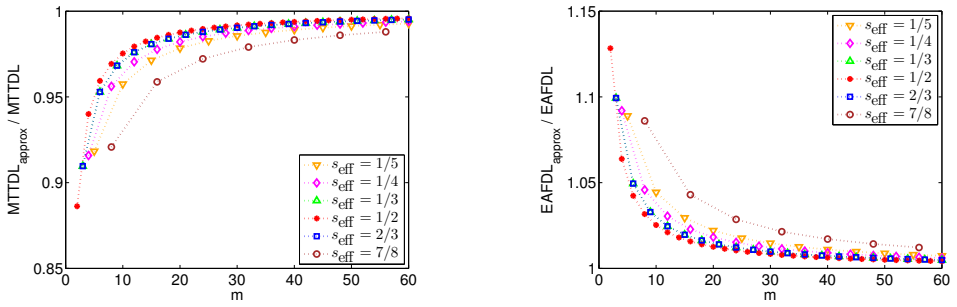
1) $m-l=1$*:* For $m-l=1$, (89) yields

$$\frac{\text{MTTDL}^{\text{declus}}}{\text{MTTDL}^{\text{clus}}} \approx \frac{m-1}{m} < 1\ . \quad (90)$$

2) $m-l=2$*:* For $m-l=2$, (89) yields

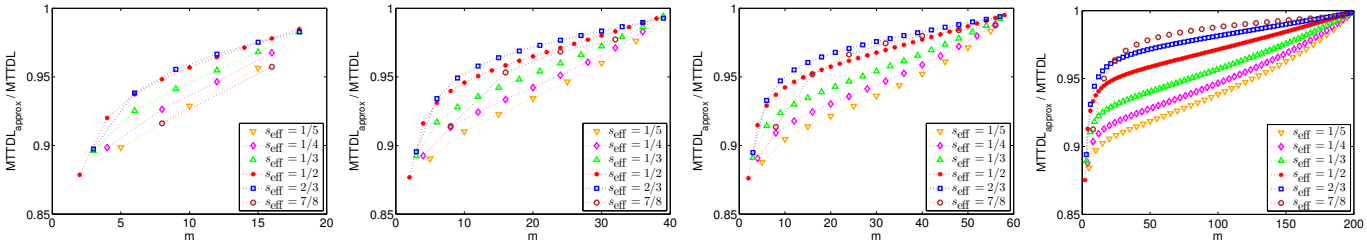
$$\frac{\text{MTTDL}^{\text{declus}}}{\text{MTTDL}^{\text{clus}}} \approx \frac{(m-2)(n-1)}{(m-1)^2} > 1\ , \ \text{ for } \ n \geq m+2\ . \quad (91)$$

Note that from (88), it holds that $2 < m \leq n/2$, which in turn implies that $n > m+2$, and therefore (91) holds.



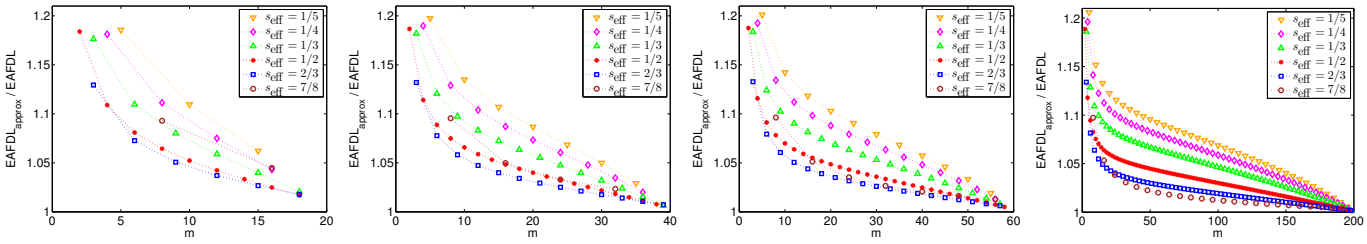
(a) $\mathrm{MTTDL}^{\mathrm{clus}}_{\mathrm{approx}}/\mathrm{MTTDL}^{\mathrm{clus}}$ (b) $\mathrm{EAFDL}^{\mathrm{clus}}_{\mathrm{approx}}/\mathrm{EAFDL}^{\mathrm{clus}}$

Figure 4. Accuracy of approximations for clustered placement vs. codeword length for $s_{\mathrm{eff}} = 1/5, 1/4, 1/3, 1/2, 2/3,$ and $7/8$.



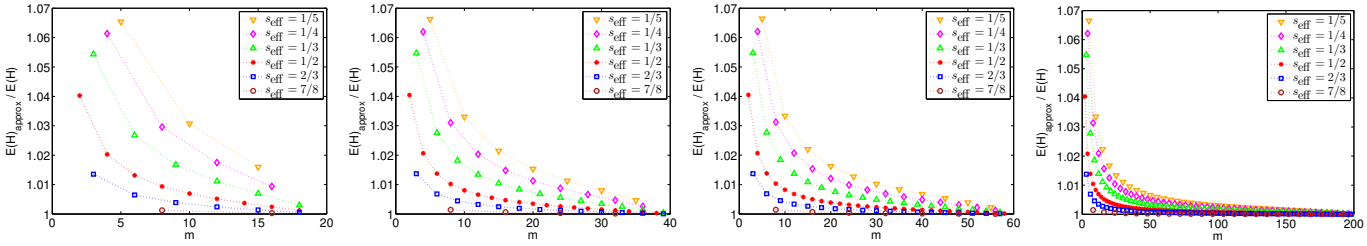
(a) $k = 20$ (b) $k = 40$ (c) $k = 60$ (d) $k = 200$

Figure 5. $\mathrm{MTTDL}^{\mathrm{sym}}_{\mathrm{approx}}/\mathrm{MTTDL}^{\mathrm{sym}}$ ratio vs. codeword length for $s_{\mathrm{eff}} = 1/5, 1/4, 1/3, 1/2, 2/3,$ and $7/8$.



(a) $k = 20$ (b) $k = 40$ (c) $k = 60$ (d) $k = 200$

Figure 6. $\mathrm{EAFDL}^{\mathrm{sym}}_{\mathrm{approx}}/\mathrm{EAFDL}^{\mathrm{sym}}$ ratio vs. codeword length for $s_{\mathrm{eff}} = 1/5, 1/4, 1/3, 1/2, 2/3,$ and $7/8$.



(a) $k = 20$ (b) $k = 40$ (c) $k = 60$ (d) $k = 200$

Figure 7. $E(H)^{\mathrm{sym}}_{\mathrm{approx}}/E(H)^{\mathrm{sym}}$ ratio vs. codeword length for $s_{\mathrm{eff}} = 1/5, 1/4, 1/3, 1/2, 2/3,$ and $7/8$.

3) $m - l \geq 3$*:* For $m - l \geq 3$, (89) can be written as follows:

$$\frac{\mathrm{MTTDL}^{\mathrm{declus}}}{\mathrm{MTTDL}^{\mathrm{clus}}} \approx \frac{m-1}{l+1} \cdots \frac{l+1}{l+1} \frac{l}{l+1} \frac{n-m+l+1}{l+1} \left(\frac{n-m+l+2}{l+2}\right)^2 \prod_{u=1}^{m-l-3} \left(\frac{n-u}{m-u}\right)^{m-l-u} . \tag{92}$$

Using (88), (92) yields

$$\begin{aligned} \frac{\mathrm{MTTDL}^{\mathrm{declus}}}{\mathrm{MTTDL}^{\mathrm{clus}}} &> \frac{l}{l+1} \frac{n-m+l+1}{l+1} \left(\frac{n-m+l+2}{l+2}\right)^2 \\ &\geq \frac{l}{l+1} \frac{l+2}{l+1} \left(\frac{l+3}{l+2}\right)^2 = \frac{l\,(l+3)^2}{(l+1)^2\,(l+2)} \\ &= \frac{2[l^2+2(l-1)+1]}{(l+1)^2\,(l+2)} + 1 > 1\,. \end{aligned} \tag{93}$$

Remark 9: From the preceding, it follows that the MTTDL is maximized by the declustered placement scheme, except in

the case of $m-l=1$, where it is maximized by the clustered placement scheme.

B. Minimizing EAFDL

From (69) and (82), it follows that

$$\frac{\text{EAFDL}^{\text{declus}}}{\text{EAFDL}^{\text{clus}}} \approx (l+1)^{m-l}\ \frac{(l-1)!}{(m-1)!}\ \prod_{u=1}^{m-l}\left(\frac{m-u}{n-u}\right)^{m-l+1-u}. \tag{94}$$

Remark 10: From (94), it follows that the placement that minimizes EAFDL does not depend on λ, b and c, nor on the rebuild time distribution.

Depending on the value of $\tilde{r}$, we consider the following two cases:

1) $m-l=1$*:* For $m-l=1$, (94) yields

$$\frac{\text{EAFDL}^{\text{declus}}}{\text{EAFDL}^{\text{clus}}} \approx \frac{m}{n-1} < 1\ ,\ \text{ for }\ n \geq m+2\ . \tag{95}$$

Note that from (88), it holds that $2 \leq m \leq n/2$, which in turn implies that $n \geq m+2$, and therefore (95) holds.

2) $m-l \geq 2$*:* For $m-l \geq 2$, (94) can be written as follows:

$$\frac{\text{EAFDL}^{\text{declus}}}{\text{EAFDL}^{\text{clus}}} \approx \frac{l+1}{m-1}\ \frac{l+1}{m-2}\ \cdots \frac{l+1}{l}\ \frac{l}{n-m+l}\ \prod_{u=1}^{m-l-1}\left(\frac{m-u}{n-u}\right)^{m-l+1-u}. \tag{96}$$

Using (88), (96) yields

$$\frac{\text{EAFDL}^{\text{declus}}}{\text{EAFDL}^{\text{clus}}} < \frac{l+1}{n-m+l} \leq \frac{l+1}{(m+1)-m+l} = 1\ . \tag{97}$$

Remark 11: From the preceding, it follows that the declustered placement scheme minimizes EAFDL for any n, m, l, λ, b, c, and rebuild time distribution.

C. Minimizing $E(H)$

From (70) and (83), and using (88), it follows that

$$\frac{E(H)^{\text{declus}}}{E(H)^{\text{clus}}} \approx \prod_{u=1}^{m-l}\frac{m-u}{n-u} < 1\ . \tag{98}$$

Remark 12: From (98), it follows that for any n, m, l, λ, b, c, and rebuild time distribution, $E(H)$ is minimized by the declustered placement scheme.

D. Synopsis

When the codeword length is smaller than the system size ($m < n$), the declustered placement scheme minimizes the expected amount of data lost when a loss occurs, independently of the device capacity c and its reliability characteristics and the mean time to failure expressed by λ, the average reserved rebuild bandwidth b and the resulting rebuild time distribution of X. Also, for $m-l=1$, the clustered placement scheme maximizes the MTTDL, but the declustered placement scheme minimizes the EAFDL. However, for $m-l \geq 2$, the declustered placement scheme maximizes the MTTDL and at the same time minimizes the EAFDL.

Note that the preceding conclusions hold under the assumption that failures are detected instantaneously, which immediately triggers the rebuild process, and the assumption that sufficient network bandwidth is available to support the parallelism of the rebuild process.

VI. Reliability Comparison

Here, we assess the relative reliability of the declustered placement, which according to Remarks 9, 11 and 12 is the optimal one, under various codeword lengths m. We perform a fair comparison by considering systems with the same amount of user data, U, stored under the same storage efficiency, s_{eff}. From (2), it follows that the number of devices n is fixed. Also, from (80), it follows that the parameter h is fixed. Using (79) to express l in terms of h and m in (81), (82), and (83), we obtain

$$\text{MTTDL}^{\text{declus}} \approx \frac{1}{n\,\lambda}\left[\frac{b}{[(1-h)m+1]\,\lambda\,c}\right]^{hm}\ (hm)!\ \frac{[E(X)]^{hm}}{E(X^{hm})}\ \prod_{u=1}^{hm}\left(\frac{n-u}{m-u}\right)^{hm-u}, \tag{99}$$

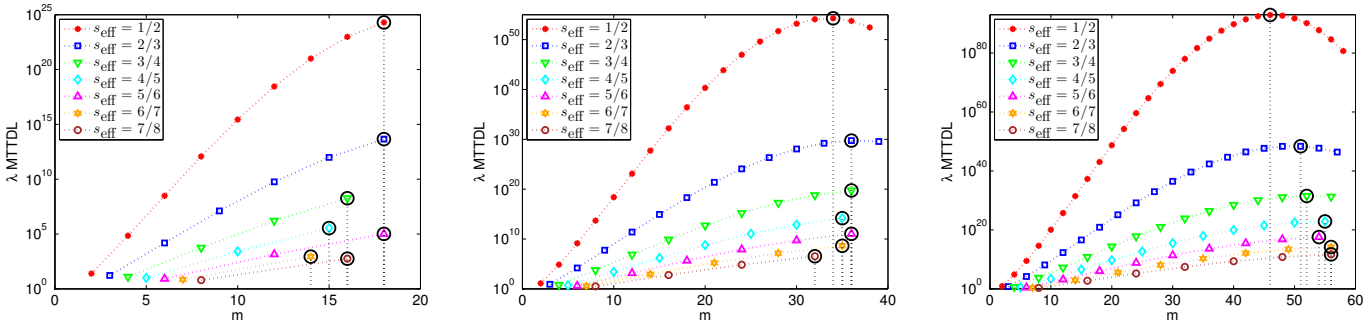
$$\text{EAFDL}^{\text{declus}} \approx \lambda\left[\frac{[(1-h)m+1]\,\lambda\,c}{b}\right]^{hm}\ \frac{m}{(hm+1)!}\ \frac{E(X^{hm})}{[E(X)]^{hm}}\ \prod_{u=1}^{hm}\left(\frac{m-u}{n-u}\right)^{hm+1-u}, \tag{100}$$

and

$$E(H)^{\text{declus}} \approx \left(\frac{(1-h)m}{hm+1}\ \prod_{u=1}^{hm}\frac{m-u}{n-u}\right)c \tag{101}$$

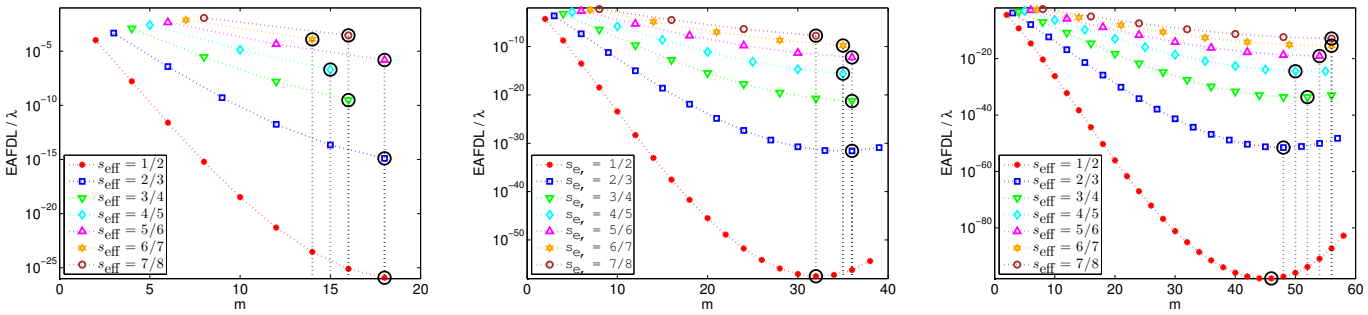
$$= \frac{(1-h)\,m\,!\,(n-1-hm)!}{(n-1)!\,(hm+1)\,((1-h)m-1))!}\,c\ . \tag{102}$$

As discussed in Section IV-A, the direct-path-approximation method yields accurate results when the storage devices are highly reliable, that is, when the ratio λ/μ of the mean rebuild time $1/\mu$ to the mean time to failure of a device $1/\lambda$ is very small. We proceed by considering systems for which it holds that $\lambda/\mu = \lambda\,c/b = 0.001$ and the rebuild time distribution is deterministic, for which it holds that $E(X^{hm}) = [E(X)]^{hm}$. The combined effect of the number of devices and the system efficiency on the normalized $\lambda\,\text{MTTDL}^{\text{declus}}$ measure is obtained by (99) and shown in Figure 8 as a function of the codeword length. The values for the storage efficiency are chosen to be fractions of the form $z/(z+1)$, $z=1,\ldots,7$, such that the first point of each of the corresponding curves is associated with the single-parity $(z, z+1)$-erasure code, and the second point of each of the corresponding curves is associated with the double-parity $(2z, 2z+2)$-erasure code. We observe that the MTTDL increases as the storage efficiency s_{eff} decreases. This is because, for a given m, decreasing s_{eff} implies decreasing l, which in turn implies increasing the parity symbols $m-l$ and consequently improving the MTTDL.



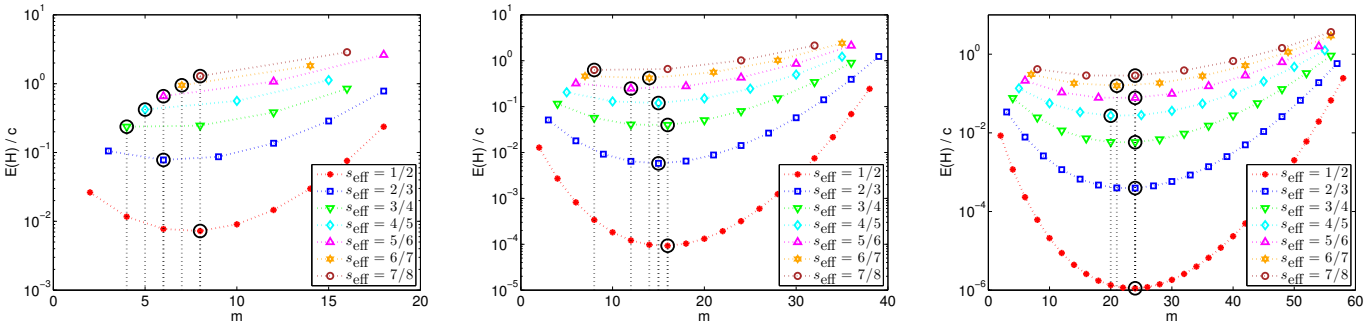
(a) $n = 20$ (b) $n = 40$ (c) $n = 60$

Figure 8. Normalized $\text{MTTDL}^{\text{declus}}$ vs. codeword length for $s_{\text{eff}} = 1/2, 2/3, 3/4, 4/5, 5/6, 6/7,$ and $7/8$; $\lambda/\mu = 0.001$ and deterministic rebuild times.



(a) $n = 20$ (b) $n = 40$ (c) $n = 60$

Figure 9. Normalized $\text{EAFDL}^{\text{declus}}$ vs. codeword length for $s_{\text{eff}} = 1/2, 2/3, 3/4, 4/5, 5/6, 6/7,$ and $7/8$; $\lambda/\mu = 0.001$ and deterministic rebuild times.



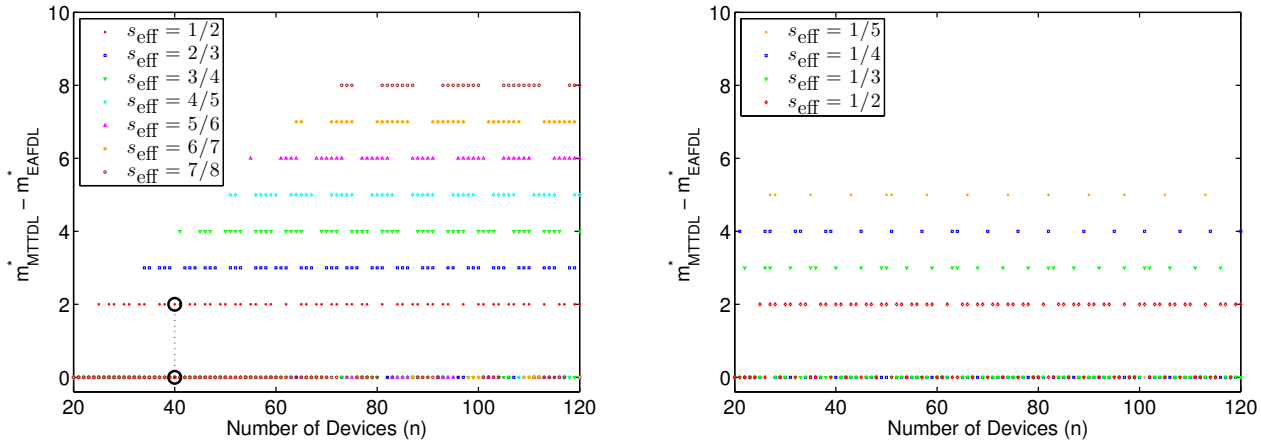
(a) $n = 20$ (b) $n = 40$ (c) $n = 60$

Figure 10. Normalized $E(H)^{\text{declus}}$ vs. codeword length for $s_{\text{eff}} = 1/2, 2/3, 3/4, 4/5, 5/6, 6/7,$ and $7/8$; $\lambda/\mu = 0.001$.

Let us now consider the single-parity codewords, which correspond to the first points of the curves. Note that, according to Remark 9 and (90), the clustered placement scheme yields larger, but of the same order, MTTDL values as the declustered placement does. Consequently, the MTTDL points for the single-parity codewords under a clustered placement scheme are slightly higher than those shown in Figure 8. As s_{eff} increases, so do m and l, which results in a decreasing MTTDL for these codewords. This is due to the fact that as m increases, there are l data symbols, that is, more data symbols associated with each parity. This is in accordance with the results presented in Figure 2 of [28]. We observe that the same applies for the double-parity codewords, which correspond to the second points of the curves.

The combined effect of the number of devices and the system efficiency on the normalized $\text{EAFDL}^{\text{declus}}/\lambda$ measure is obtained by (100) and shown in Figure 9 as a function of the codeword length. We observe that the EAFDL increases as the storage efficiency s_{eff} increases. Also, as s_{eff} increases, the EAFDL for the single-parity codewords, which correspond to the first points of the curves, also increases. We observe that the same applies for the double-parity codewords, which correspond to the second points of the curves.

The combined effect of the number of devices and the system efficiency on the normalized $E(H)^{\text{declus}}/c$ measure is obtained by (101) and shown in Figure 10 as a function of the codeword length. We observe that $E(H)$ increases as the storage efficiency s_{eff} increases. Also, as s_{eff} increases, the $E(H)$ for the single-parity codewords, which correspond to the first points of the curves, increases as well. We observe that the same applies for the double-parity codewords, which correspond to the second points of the curves.



(a) $s_{\text{eff}} = 1/2, 2/3, 3/4, 4/5, 5/6, 6/7,$ and $7/8$

(b) $s_{\text{eff}} = 1/5, 1/4, 1/3,$ and $1/2$

Figure 11. The difference between m^*_{MTTDL} and m^*_{EAFDL} vs. the number of devices; $\lambda/\mu = 0.001$ and deterministic rebuild times.

We now proceed to identify the optimal codeword length, m^*, that maximizes the MTTDL or minimizes the EAFDL and $E(H)$ for a given storage efficiency. Note that we only consider declustered placements with $m < n$, but not the clustered placement with $m = n$. The optimal codeword length is dictated by two opposing effects on reliability. On the one hand, larger values of m imply that codewords can tolerate more device failures, but on the other hand result in a higher exposure degree to failure as each of the codewords is spread across a larger number of devices. In Figures 8, 9, and 10, the optimal values, m^*, are indicated by the circles, and the corresponding codeword lengths are indicated by the vertical dotted lines. Regarding MTTDL and EAFDL, we observe that for small values of n, it holds that $m^* \cong n$, whereas for large values of n it holds that $m^* < n$. It turns out that for $n \geq 60$, the clustered placement scheme with $m = n$ does not result in improved reliability. However, for smaller values of n, that is, for $n < 60$, the clustered placement scheme can improve reliability. For instance, for $n = 20$ and $s_{\text{eff}} = 4/5$, the MTTDL is maximized and the EAFDL is minimized by the clustered placement scheme with $m = n$. By comparing Figures 8 and 9, we deduce that in general the optimal codeword lengths m^*_{MTTDL} (for MTTDL) and m^*_{EAFDL} (for EAFDL) are similar and for some values of n even identical. They are, however, significantly larger than those that minimize the $E(H)$, which are shown in Figure 10.

Figure 11 shows the difference between the optimal codeword lengths for MTTDL and EAFDL. It demonstrates that the optimal codeword length for MTTDL is always greater than or equal to that for EAFDL, with the difference being equal either to $z+1$, the denominator of the storage efficiency fraction, or to zero. This implies that the optimal codeword lengths m^*_{EAFDL} for EAFDL are either equal to or slightly lower than and adjacent to the optimal codeword lengths m^*_{MTTDL} for MTTDL. For example, in the case of $n = 40$ and $s_{\text{eff}} = 1/2$, Figure 8(b) shows that the maximum value of MTTDL is achieved when the codeword length m is equal to 34, which implies that $m^*_{\text{MTTDL}} = 34$. Also, Figure 9(b) shows that the minimum value of EAFDL is achieved when the codeword length m is equal to 32, which implies that $m^*_{\text{EAFDL}} = 32$. The value of 32 is adjacent to 34 because when $s_{\text{eff}} = 1/2$, m cannot be equal to 33. Consequently, the difference of the optimal codeword lengths for EAFDL and MTTDL is given by 34 – 32 = 2, indicated by a circle in Figure 11. Similarly, for $n = 40$ and $s_{\text{eff}} = 2/3$, Figures 8(b) and 9(b) show that both the optimal MTTDL and the optimal EAFDL are obtained when the codeword length is equal to 36, that is, $m^*_{\text{MTTDL}} = m^*_{\text{EAFDL}} = 36$. In this case, the difference of the optimal codeword lengths for EAFDL and MTTDL is equal to zero, indicated by a circle in Figure 11.

To investigate the behavior of the optimal codeword length, m^*, as the storage system size, n, increases, we proceed by considering the normalized optimal codeword length r^*, namely, the ratio of m^* to n:

$$r^* \triangleq \frac{m^*}{n} \,. \tag{103}$$

The r^* values for various storage efficiencies and for the MTTDL and EAFDL metrics are shown in Figure 12. From the preceding, it follows that the difference $r^*_{\text{MTTDL}} - r^*_{\text{EAFDL}}$ of the r^* values for the two metrics is bounded above by $(z+1)/n$, which approaches zero as n increases. Thus, as n increases, the difference $r^*_{\text{MTTDL}} - r^*_{\text{EAFDL}}$ also approaches zero.

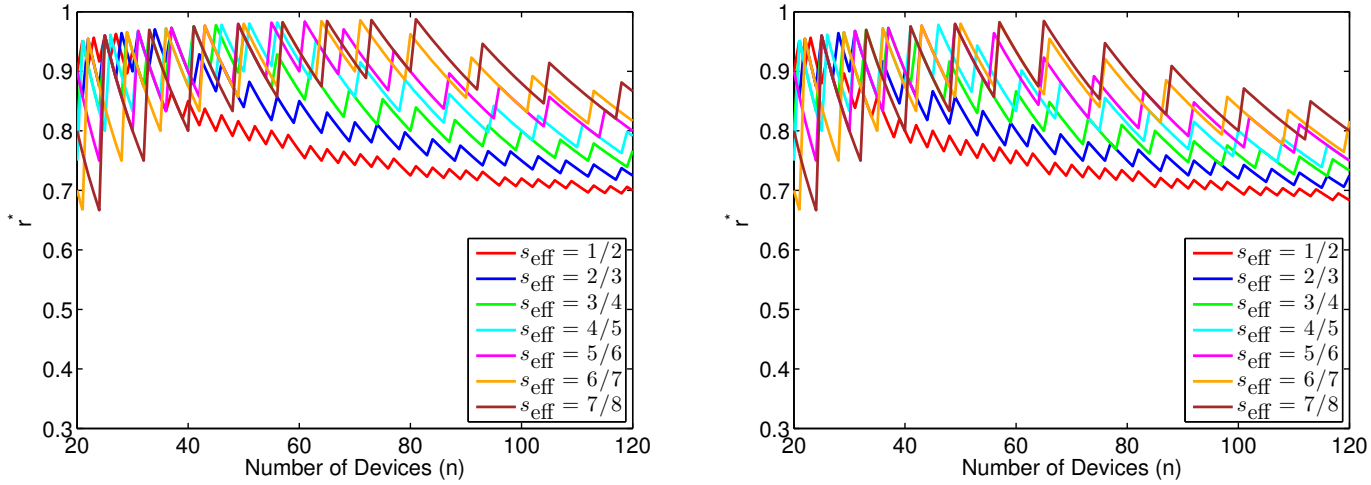
The r^* values for the MTTDL and EAFDL metrics for various values of the storage efficiency s_{eff} and for large values of n are shown in Figures 13 and 14. It turns out that it always holds that $r^*_{\text{EAFDL}} \leq r^*_{\text{MTTDL}}$ or, equivalently, $m^*_{\text{EAFDL}} \leq m^*_{\text{MTTDL}}$. We observe that, as n increases, the r^* values tend to decrease. In particular, for a given storage efficiency and as n increases, the r^* values for MTTDL and EAFDL approach a common value, denoted by r^*_∞ and indicated by a small bullet. The r^*_∞ value depends only on s_{eff} and is given by the following proposition.

Proposition 11: As n increases, the r^* values for MTTDL and EAFDL approach r^*_∞ that satisfies the following equation:

$$Q(h, r^*_\infty) = 0 \,, \tag{104}$$

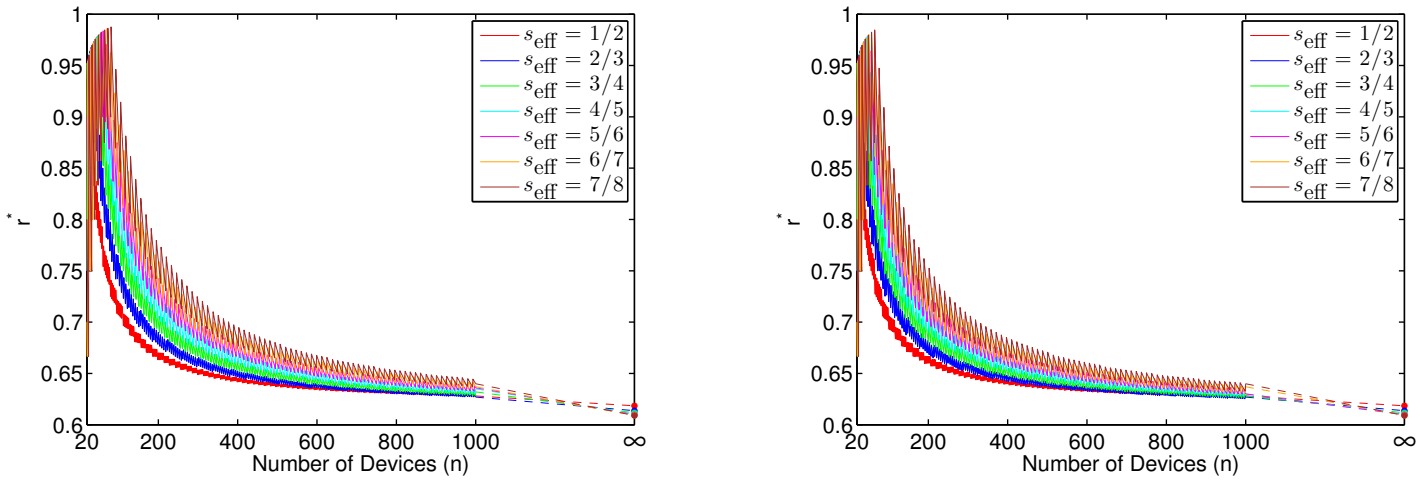
where

$$Q(h,x) \triangleq hx + \log\left([(1-h)^{(1-h)^2} x^{h^2}]^x (1-hx)^{h(1-hx)}\right) . \tag{105}$$



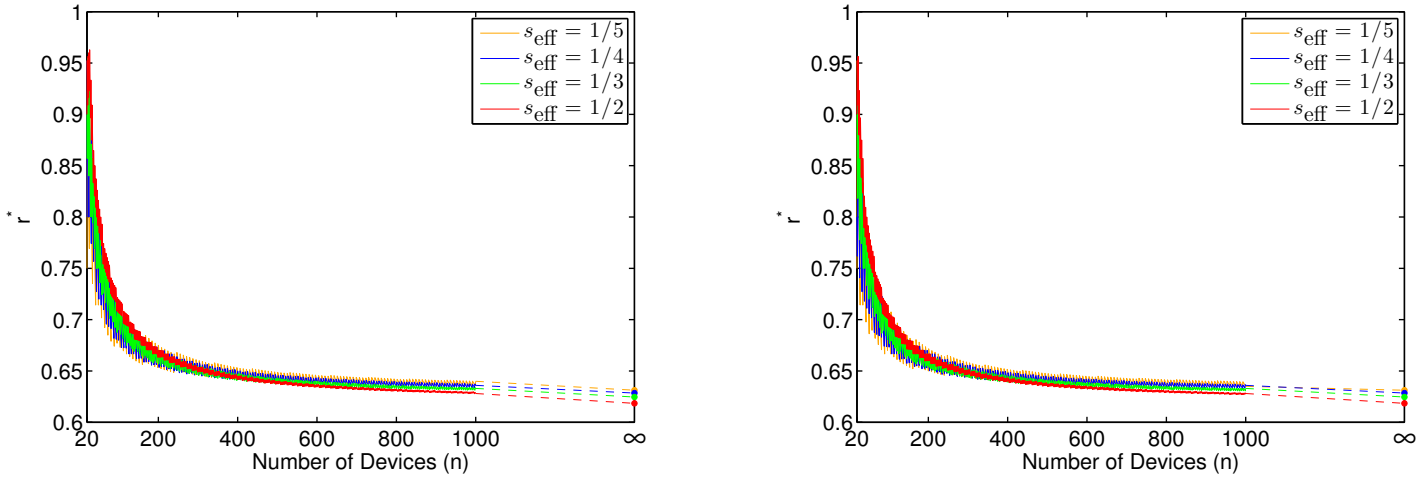
(a) MTTDL (b) EAFDL

Figure 12. r^* vs. number of devices for $s_{\text{eff}} = 1/2, 2/3, 3/4, 4/5, 5/6, 6/7,$ and $7/8$; $\lambda/\mu = 0.001$ and deterministic rebuild times.



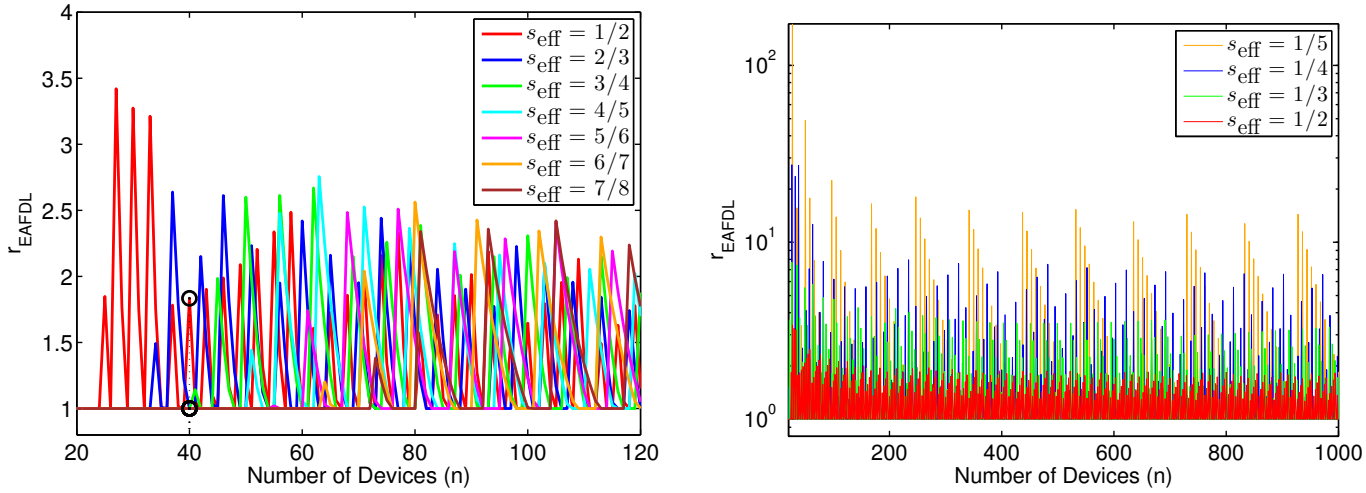
(a) MTTDL (b) EAFDL

Figure 13. r^* vs. number of devices $n \to \infty$, $s_{\text{eff}} = 1/2, 2/3, 3/4, 4/5, 5/6, 6/7,$ and $7/8$; $\lambda/\mu = 0.001$ and deterministic rebuild times.



(a) MTTDL (b) EAFDL

Figure 14. r^* vs. number of devices $n \to \infty$, $s_{\text{eff}} = 1/5, 1/4, 1/3,$ and $1/2$; $\lambda/\mu = 0.001$ and deterministic rebuild times.



(a) $s_{\text{eff}} = 1/2, 2/3, 3/4, 4/5, 5/6, 6/7,$ and $7/8$

(b) $s_{\text{eff}} = 1/5, 1/4, 1/3,$ and $1/2$

Figure 16. The EAFDL efficiency ratio r_{EAFDL} vs. number of devices; $\lambda/\mu = 0.001$ and deterministic rebuild times.

TABLE III. r^*_∞ VALUES FOR VARIOUS s_{eff}

s_{eff}			r^*_∞	
			MTTDL and EAFDL	$E(H)$
0	=	0	0.648419	0.5
10^{-4}	=	0.0001	0.648404	0.499795
10^{-3}	=	0.001	0.648265	0.498520
10^{-2}	=	0.01	0.646985	0.490770
10^{-1}	=	0.1	0.637940	0.456298
1/8	=	0.125	0.636043	0.450268
1/7	=	0.142857	0.634788	0.446383
1/6	=	0.166667	0.633224	0.441637
1/5	=	0.2	0.631212	0.435664
1/4	=	0.25	0.628500	0.427826
1/3	=	0.333333	0.624638	0.416889
1/2	=	0.5	0.618499	0.4
2/3	=	0.666667	0.613720	0.387097
3/4	=	0.75	0.611679	0.381625
4/5	=	0.8	0.610543	0.378586
5/6	=	0.833333	0.609818	0.376650
6/7	=	0.857143	0.609316	0.375307
7/8	=	0.875	0.608946	0.374322
$1-10^{-1}$	=	0.9	0.608440	0.372971
$1-10^{-2}$	=	0.99	0.606713	0.368368
$1-10^{-3}$	=	0.999	0.606549	0.367928
$1-10^{-4}$	=	0.9999	0.606532	0.367884
1	=	1	0.606531 = $1/\sqrt{e}$	0.367879 = $1/e$

Proof: See Appendix F. ■

The r^*_∞ values corresponding to the MTTDL and EAFDL metrics and to various storage efficiencies are listed in Table III. Note that the r^*_∞ values are in the interval $[e^{-1/2} = 0.606, 0.648]$ and decrease as the storage efficiency s_{eff} increases. In contrast, for small values of n, the r^* values increase as the storage efficiency increases, as shown in Figure 13. For example, for small n, the r^* values corresponding to $s_{\text{eff}} = 1/2$ are smaller than those corresponding to $s_{\text{eff}} = 2/3$. However, for large values of n this is reversed, and for the MTTDL, the first instance that this occurs is for $n = 619$, as shown in Figure 15, with the r^* values being equal to 0.637 and 0.635 (indicated by the circle), respectively. Therefore, in this case, the optimal codeword lengths m^* are equal to 394 and 393, respectively.

Next we examine the increase of the EAFDL metric if

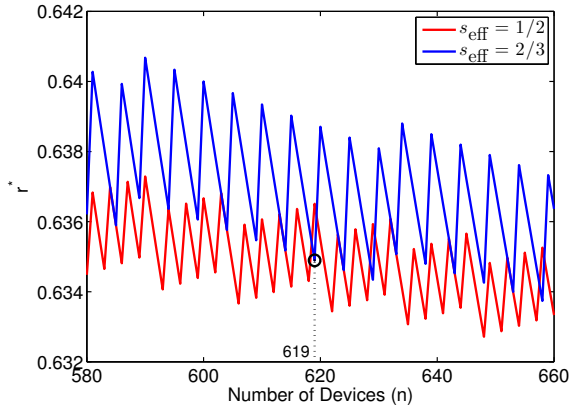
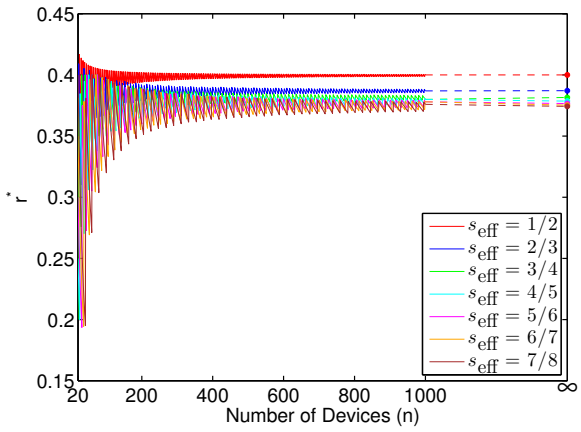


Figure 15. r^* for MTTDL vs. number of devices for $s_{\text{eff}} = 1/2, 2/3$; $\lambda/\mu = 0.001$ and deterministic rebuild times.

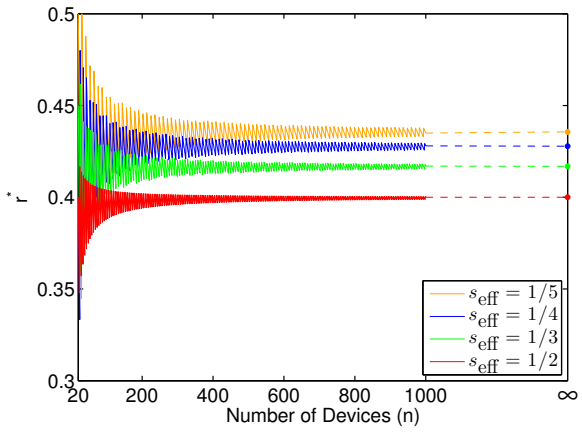
instead of the optimal codeword lengths m^*_{EAFDL}, we use the codeword lengths m^*_{MTTDL} that optimize the MTTDL metric. From the preceding, it follows that m^*_{MTTDL} is either equal to m^*_{EAFDL} or adjacent to it, that is, $m^*_{\text{MTTDL}} = m^*_{\text{EAFDL}} + z + 1$. We define the EAFDL *efficiency ratio*, r_{EAFDL}, as the ratio of EAFDL(m^*_{MTTDL}) to EAFDL(m^*_{EAFDL}), that is,

$$r_{\text{EAFDL}} \triangleq \frac{\text{EAFDL}(m^*_{\text{MTTDL}})}{\text{EAFDL}(m^*_{\text{EAFDL}})}\,, \tag{106}$$

where EAFDL(m) denotes the EAFDL corresponding to a codeword length m. In the case of $n = 40$ and $s_{\text{eff}} = 1/2$, from the preceding and according to Figure 9(b), it holds that EAFDL(m^*_{EAFDL}) = EAFDL(32) = 3.08×10^{-58} and EAFDL(m^*_{MTTDL}) = EAFDL(34) = 5.66×10^{-58}, which yields an EAFDL efficiency ratio r_{EAFDL} of 5.66/3.08 = 1.84. This is indicated by a circle in Figure 16(a), which shows the EAFDL efficiency ratio as a function of n. Similarly, in the case of $n = 40$ and $s_{\text{eff}} = 2/3$, from the preceding, it holds that $m^*_{\text{MTTDL}} = m^*_{\text{EAFDL}} = 36$, which implies that $r_{\text{EAFDL}} = 1$, indicated by a circle in Figure 16(a). We observe that for the storage efficiencies considered and as n increases,



(a) $s_{\text{eff}} = 1/2, 2/3, 3/4, 4/5, 5/6, 6/7,$ and $7/8$



(b) $s_{\text{eff}} = 1/5, 1/4, 1/3,$ and $1/2$

Figure 18. r^* for $E(H)$ vs. number of devices $n \to \infty$; $\lambda/\mu = 0.001$.

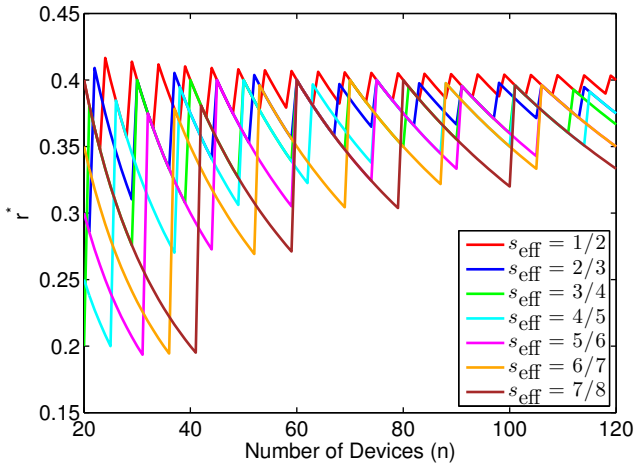


Figure 17. r^* for $E(H)$ vs. number of devices for $s_{\text{eff}} = 1/2, 2/3, 3/4, 4/5, 5/6, 6/7,$ and $7/8$; $\lambda/\mu = 0.001$.

the EAFDL efficiency ratios follow a periodic pattern and are always less than a factor of four. This implies that using codewords of length m^*_{MTTDL} yields the maximum possible (optimal) MTTDL and also an EAFDL that is either the optimal one or of the same order as the optimal one. Also, as the storage efficiency decreases, the EAFDL efficiency ratio r_{EAFDL} increases, as shown in Figure 16(b). For any given storage efficiency, r_{EAFDL} follows a periodic pattern and for $s_{\text{eff}} \geq 1/4 = 0.25$, r_{EAFDL} is always less than a factor of 10. Consequently, using codewords of length m^*_{MTTDL} yields an EAFDL that is either the optimal or at most one order of magnitude higher than the optimal one.

Next, we compare the r^* values for the MTTDL and EAFDL metrics shown in Figure 12 with those for the $E(H)$ metric shown in Figure 17. Clearly, the optimal codeword lengths for MTTDL and EAFDL are significantly larger than those that minimize $E(H)$. The r^* values for the $E(H)$ metric for various values of the storage efficiency s_{eff} and for large values of n are shown in Figure 18. The figure indicates that, as n increases, the r^* values oscillate and approach a value denoted by r^*_∞. The r^*_∞ values (indicated by the small bullets) are given by the following proposition,

Proposition 12: As n increases, the r^* values for $E(H)$ approach r^*_∞ given by

$$r^*_\infty = \frac{1}{h + (1-h)^{-\frac{1-h}{h}}}, \tag{107}$$

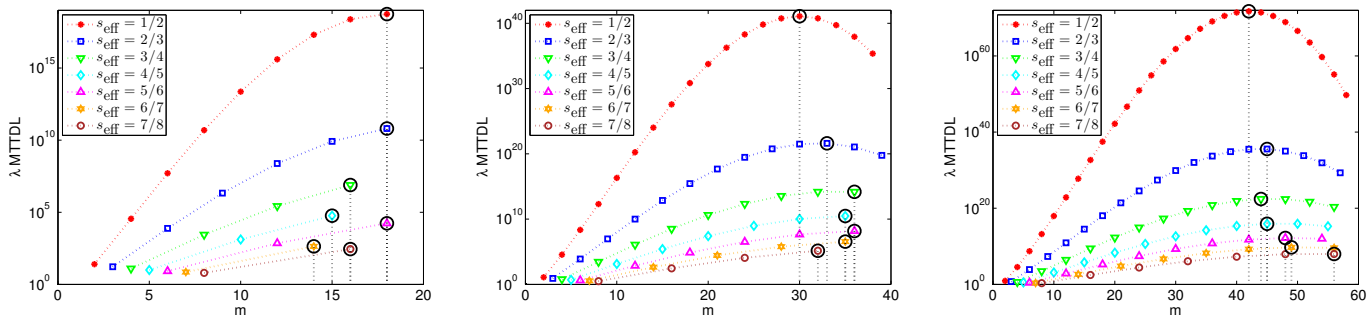
where h is given by (60).

Proof: See Appendix G. ■

The r^*_∞ values corresponding to the $E(H)$ metric and to various storage efficiencies are listed in Table III. Note that the r^*_∞ values are in the interval $[e^{-1} = 0.368, 0.5]$ and decrease as the storage efficiency s_{eff} increases. By inspecting Figures 13, 14, and 18, it is evident that also in this case the optimal codeword lengths for MTTDL and EAFDL are significantly larger than those that minimize $E(H)$.

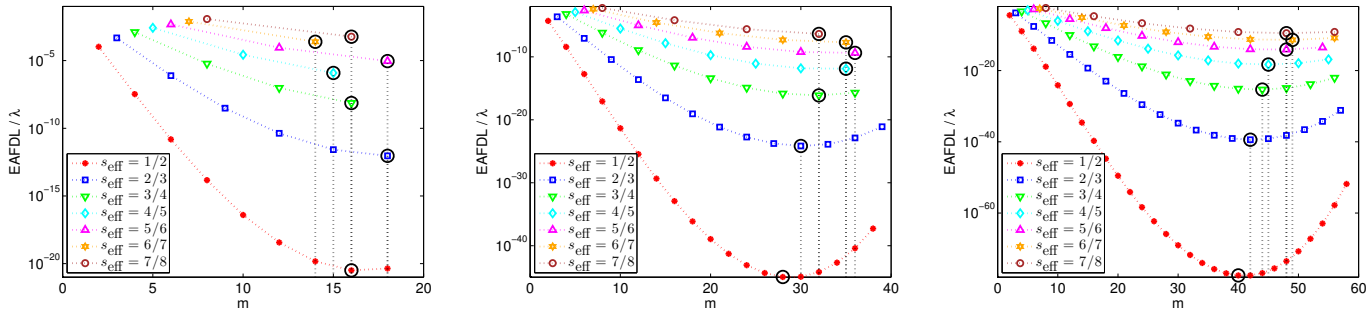
Next, we consider a system where the distribution of the rebuild time X is exponential, for which it holds that $E(X^{hm}) = (hm)!\,[E(X)]^{hm}$. According to Remark 2, this only affects the MTTDL and EAFDL metrics, but not the $E(H)$ metric. The combined effect of the number of devices and the system efficiency on the normalized $\lambda\,\text{MTTDL}^{\text{declus}}$ measure is obtained by (99) and shown in Figure 19 as a function of the codeword length. Similarly to the case of deterministic rebuild times, we observe that the MTTDL increases as the storage efficiency s_{eff} decreases. Also, as s_{eff} increases, the MTTDL for the single-parity codewords, which correspond to the first points of the curves, decreases. We observe that the same applies for the double-parity codewords, which correspond to the second points of the curves.

The combined effect of the number of devices and the system efficiency on the normalized $\text{EAFDL}^{\text{declus}}/\lambda$ measure is obtained by (100) and shown in Figure 20 as a function of the codeword length. Similarly to the case of deterministic rebuild times, we observe that the EAFDL increases as the storage efficiency s_{eff} increases. Also, as s_{eff} increases, the EAFDL for the single-parity codewords, which correspond to the first points of the curves, also increases. We observe that the same applies for the double-parity codewords, which correspond to the second points of the curves.



(a) $n = 20$ (b) $n = 40$ (c) $n = 60$

Figure 19. Normalized $\text{MTTDL}^{\text{declus}}$ vs. codeword length for $s_{\text{eff}} = 1/2, 2/3, 3/4, 4/5, 5/6, 6/7,$ and $7/8$; $\lambda/\mu = 0.001$ and exponential rebuild times.



(a) $n = 20$ (b) $n = 40$ (c) $n = 60$

Figure 20. Normalized $\text{EAFDL}^{\text{declus}}$ vs. codeword length for $s_{\text{eff}} = 1/2, 2/3, 3/4, 4/5, 5/6, 6/7,$ and $7/8$; $\lambda/\mu = 0.001$ and exponential rebuild times.

The optimal codeword lengths, m^*, that maximize the MTTDL or minimize the EAFDL are indicated by the circles and the corresponding vertical dotted lines. The observations regarding the optimal codeword lengths made in the case of deterministic rebuild times also apply here. Note also that, according to Remark 7, $E(H)$ does not depend on the rebuild times, and therefore the optimal codeword lengths that minimize $E(H)$ are those shown in Figure 17 for the case of deterministic rebuild times.

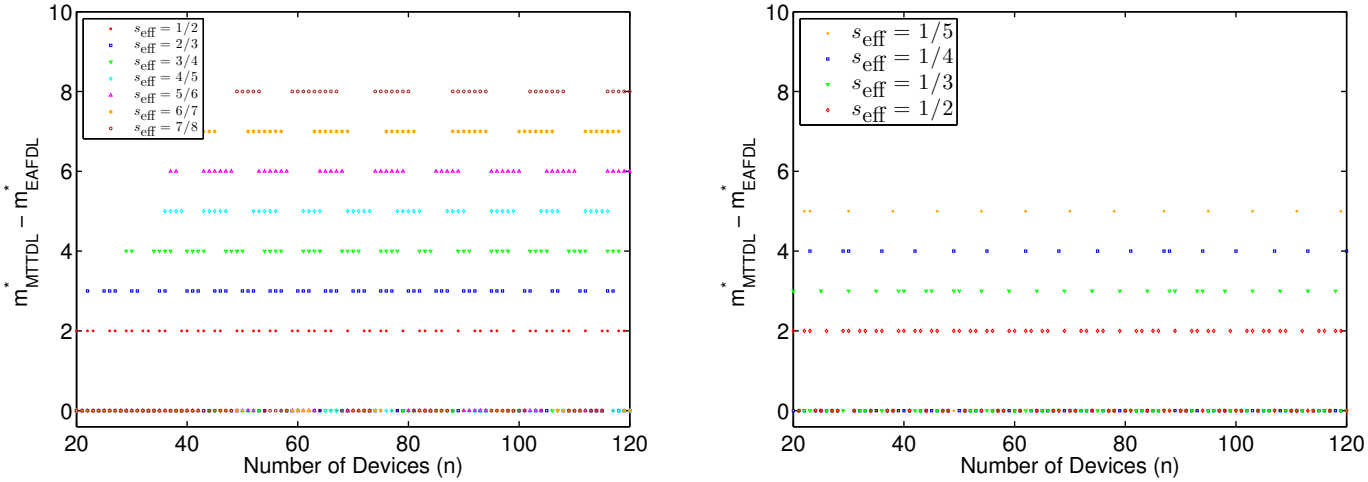
Similarly to the case of deterministic rebuild times, the optimal codeword lengths m^*_{EAFDL} for EAFDL are either equal to or slightly lower than and adjacent to the optimal codeword lengths m^*_{MTTDL} for MTTDL, as demonstrated in Figure 21. The r^* values for the MTTDL and EAFDL metrics for various storage efficiencies are shown in Figure 22. In Appendix F, it is proved that as n increases, and for any storage efficiency, the r^* values for MTTDL and EAFDL approach a common value that is the same as the r^*_∞ value obtained in the case of deterministic rebuild times, which depends on s_{eff} and is listed in Table III.

The EAFDL efficiency ratios r_{EAFDL} as a function of n for various storage efficiencies are shown in Figure 23. We observe that for the storage efficiencies considered and as n increases, the EAFDL efficiency ratios follow a periodic pattern, and for $s_{\text{eff}} \geq 1/4 = 0.25$, they are always less than a factor of 10. By inspecting Figures 16 and 23, we observe that the r_{EAFDL} ratios in the case of exponential rebuild times are smaller than those in the case of deterministic rebuild times.

Figures 24 and 25 show the ratio of the optimal codeword length, m^*_{exp}, for the exponential distribution to the optimal codeword length, m^*_{det}, for the deterministic distribution for various storage efficiencies. We observe that this ratio never exceeds one and approaches one as n increases. This implies that the optimal codeword length for the exponential distribution is in general smaller than the optimal codeword length for the deterministic distribution. This can be intuitively explained as follows. As previously mentioned, larger values of m result in a higher exposure degree to failure as each of the codewords is spread across a larger number of devices. The variation of exponentially distributed rebuild times results in increased vulnerability windows and therefore worse reliability. To reduce the exposure degree to failures, codewords should be spread across a smaller number of devices, which implies a smaller optimal codeword length.

VII. Conclusions

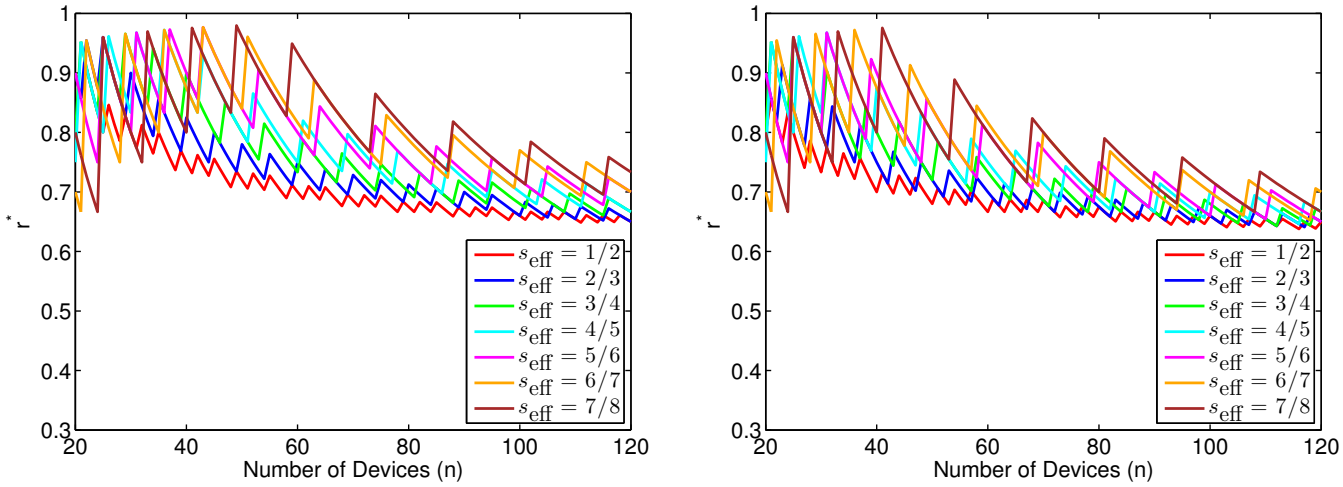
We considered the Mean Time to Data Loss (MTTDL) and the Expected Annual Fraction of Data Loss (EAFDL) reliability metrics of storage systems using advanced erasure codes. A methodology was presented for deriving the two metrics analytically. Closed-form expressions capturing the effect of various system parameters were obtained for arbitrary rebuild time distributions and for the symmetric, clustered, and declustered data placement schemes. We established that the declustered placement scheme offers superior reliability in terms of both metrics. Subsequently, a thorough comparison of the reliability achieved by the declustered placement scheme under various codeword configurations was conducted. The results obtained show that the optimal codeword lengths for MTTDL and EAFDL are similar and, as the system size grows, they are about 60% of the storage system size.



(a) $s_{\text{eff}} = 1/2, 2/3, 3/4, 4/5, 5/6, 6/7,$ and $7/8$

(b) $s_{\text{eff}} = 1/5, 1/4, 1/3,$ and $1/2$

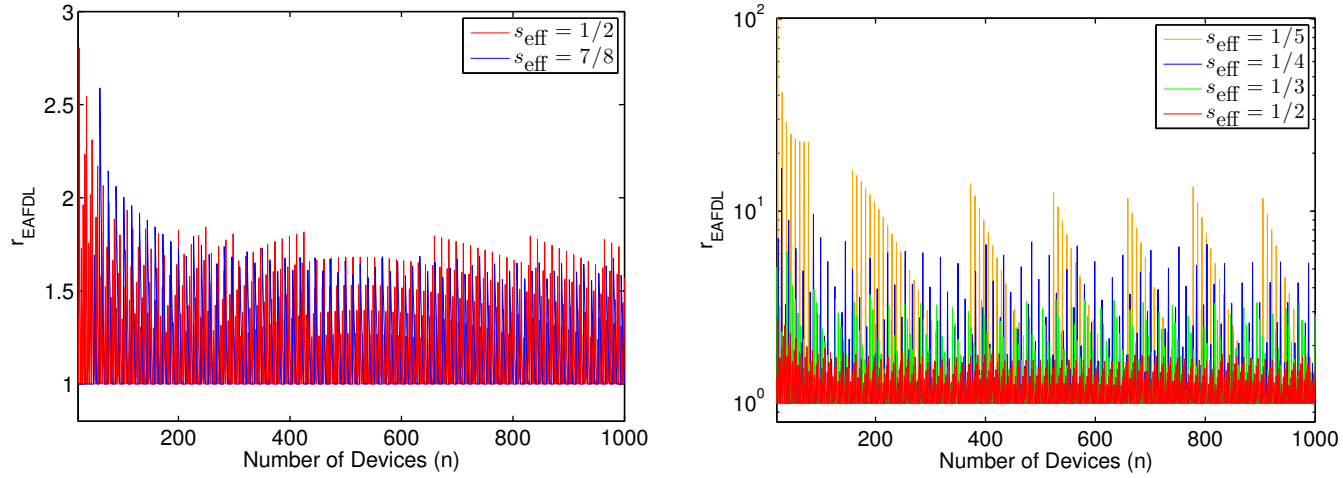
Figure 21. The difference between m^*_{MTTDL} and m^*_{EAFDL} vs. number of devices; $\lambda/\mu = 0.001$ and exponential rebuild times.



(a) MTTDL

(b) EAFDL

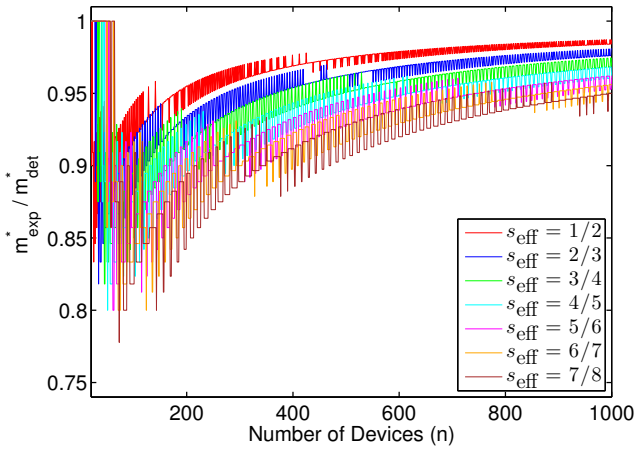
Figure 22. r^* vs. number of devices for $s_{\text{eff}} = 1/2, 2/3, 3/4, 4/5, 5/6, 6/7,$ and $7/8$; $\lambda/\mu = 0.001$ and exponential rebuild times.



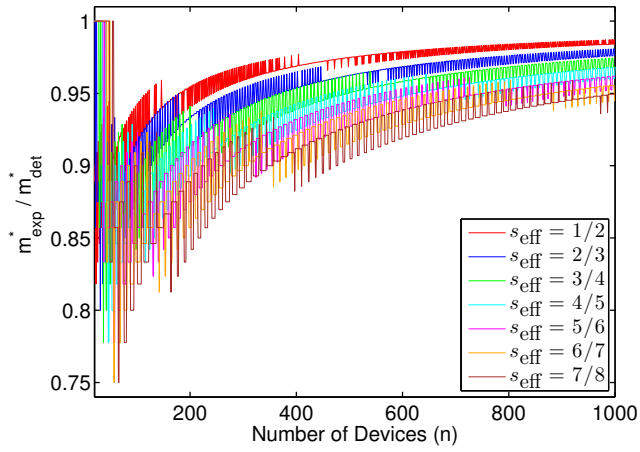
(a) $s_{\text{eff}} = 1/2$ and $7/8$

(b) $s_{\text{eff}} = 1/5, 1/4, 1/3,$ and $1/2$

Figure 23. The EAFDL efficiency ratio r_{EAFDL} vs. number of devices; $\lambda/\mu = 0.001$ and exponential rebuild times.

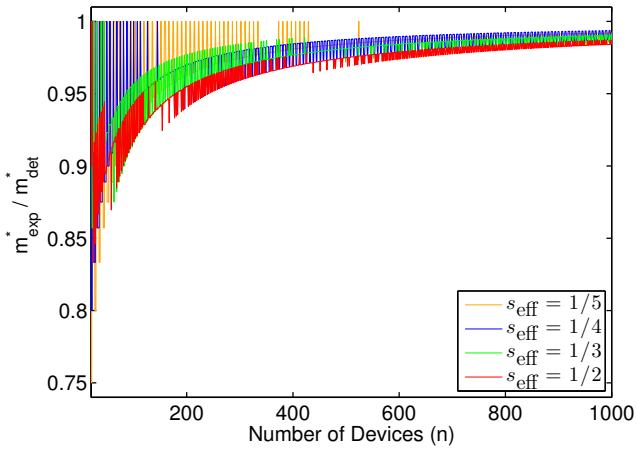


(a) MTTDL

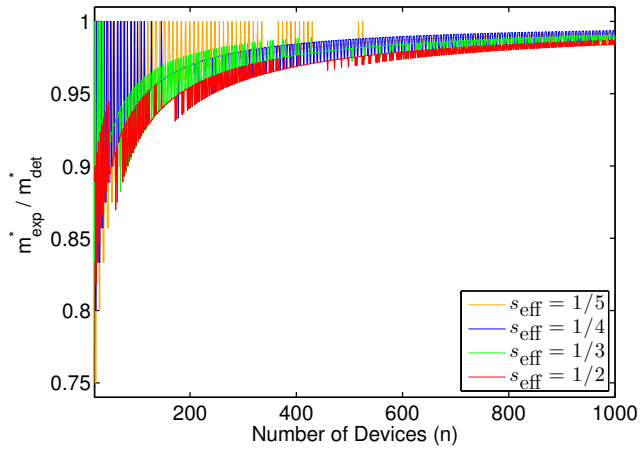


(b) EAFDL

Figure 24. Ratio m^*_{exp} to m^*_{det} vs. number of devices for $s_{\text{eff}} = 1/2, 2/3, 3/4, 4/5, 5/6, 6/7,$ and $7/8$; $\lambda/\mu = 0.001$.



(a) MTTDL



(b) EAFDL

Figure 25. Ratio m^*_{exp} to m^*_{det} vs. number of devices for $s_{\text{eff}} = 1/5, 1/4, 1/3,$ and $1/2$; $\lambda/\mu = 0.001$.

Extending the methodology developed to derive the reliability of erasure coded systems under network rebuild bandwidth limitations and in the presence of unrecoverable latent errors are subjects of further investigation. Also, owing to the parallelism of the rebuild process, the model considered yields very small rebuild times for large system sizes. To take into account the fact that the rebuild times cannot be smaller than the actual failure detection times requires a more sophisticated modeling effort, which is also part of future work.

Appendix A
Estimation of P_{DL}

Proof of Proposition 2.

Consider the direct path $1 \to 2 \to \cdots \to \tilde{r}$ of successive transitions from exposure level 1 to $\tilde{r}$. For ease of reading, we denote the successive transitions from exposure level u to $\tilde{r}$ by $u \to \tilde{r}$. We first evaluate $P_{\text{DL}}(R_1)$, the probability of data loss conditioned on the rebuild time R_1. From (19), and using the fact that α_u does not depend on $R_1, \alpha_1, \cdots, \alpha_{u-1}$, it follows that

$$
\begin{aligned}
P_{\text{DL}}(R_1) \;&\approx\; P_{1\to\tilde{r}}(R_1) \\
&= P_{1\to 2}(R_1) P_{2\to\tilde{r}}(R_1) \\
&= P_{1\to 2}(R_1) E_{\alpha_1|R_1}[P_{2\to\tilde{r}}(R_1,\alpha_1)] \\
&= P_{1\to 2}(R_1) E_{\alpha_1}[P_{2\to 3}(R_1,\alpha_1) P_{3\to\tilde{r}}(R_1,\alpha_1)] \\
&= P_{1\to 2}(R_1) E_{\alpha_1}[P_{2\to 3}(R_1,\alpha_1) E_{\alpha_2|R_1,\alpha_1}[P_{3\to\tilde{r}}(R_1,\alpha_1,\alpha_2)]] \\
&= \cdots \\
&= P_{1\to 2}(R_1) E_{\alpha_1}[P_{2\to 3}(R_1,\vec{\alpha}_1) E_{\alpha_2}[P_{3\to 4}(R_1,\vec{\alpha}_2) \cdots \\
&\qquad\qquad \cdots E_{\alpha_{\tilde{r}-2}}[P_{\tilde{r}-1\to\tilde{r}}(R_1,\vec{\alpha}_{\tilde{r}-2})] \cdots] \\
&= E_{\vec{\alpha}_{\tilde{r}-2}}[P_{1\to 2}(R_1) P_{2\to 3}(R_1,\vec{\alpha}_1) \cdots P_{\tilde{r}-1\to\tilde{r}}(R_1,\vec{\alpha}_{\tilde{r}-2})] \\
&= E_{\vec{\alpha}_{\tilde{r}-2}}\left[\prod_{u=1}^{\tilde{r}-1} P_{u\to u+1}(R_1,\vec{\alpha}_{u-1})\right] \\
&= E_{\vec{\alpha}_{\tilde{r}-2}}[P_{\text{DL}}(R_1,\vec{\alpha}_{\tilde{r}-2})]\;, \qquad (108)
\end{aligned}
$$

where

$$
P_{\text{DL}}(R_1,\vec{\alpha}_{\tilde{r}-2}) \;\triangleq\; \prod_{u=1}^{\tilde{r}-1} P_{u\to u+1}(R_1,\vec{\alpha}_{u-1})\;, \qquad (109)
$$

with

$$P_{1\to 2}(R_1, \vec{\alpha}_0) \triangleq P_{1\to 2}(R_1) \; . \tag{110}$$

Substituting (45) into (109), and using (44) and (110), yields

$$P_{\text{DL}}(R_1, \vec{\alpha}_{\tilde{r}-2}) \approx (\lambda\, b_1\, R_1)^{\tilde{r}-1} \prod_{u=1}^{\tilde{r}-1} \frac{\tilde{n}_u}{b_u} \, (V_u\, \alpha_u)^{\tilde{r}-1-u} \; . \tag{111}$$

Unconditioning (111) on $\vec{\alpha}_{\tilde{r}-2}$, and given that the elements of $\vec{\alpha}_{\tilde{r}-2}$ are independent random variables approximately distributed according to (24) such that $E(\alpha_u^k) \approx 1/(k+1)$, (108) yields

$$P_{\text{DL}}(R_1) \approx (\lambda\, b_1\, R_1)^{\tilde{r}-1} \, \frac{1}{(\tilde{r}-1)!} \prod_{u=1}^{\tilde{r}-1} \frac{\tilde{n}_u}{b_u} \, V_u^{\tilde{r}-1-u} \; . \tag{112}$$

The probability of data loss P_{DL} is obtained by unconditioning $P_{\text{DL}}(R_1)$ on R_1, that is,

$$P_{\text{DL}} = E[P_{\text{DL}}(R_1)] \; . \tag{113}$$

Unconditioning (112) on R_1 using (10) and (34), (113) yields (46). ■

Appendix B
Estimation of $E(Q)$

Proof of Proposition 3.

We first evaluate $E(Q|R_1)$, the expected amount of data lost conditioned on the rebuild time R_1. From (21), and considering the direct path $1 \to 2 \to \cdots \to \tilde{r}$ of successive transitions from exposure level 1 to $\tilde{r}$, and using the fact that α_u does not depend on $R_1, \alpha_1, \cdots, \alpha_{u-1}$, it follows that

$$\begin{aligned}
E(Q|R_1) \; &\approx \; P_{1\to 2}(R_1) E(Q|R_1, 1\to 2) \\
&= P_{1\to 2}(R_1) E_{\alpha_1|R_1}[E(Q|R_1, \alpha_1)] \\
&= P_{1\to 2}(R_1) E_{\alpha_1}[P_{2\to 3}(R_1, \alpha_1) E(Q|R_1, \alpha_1, 2\to 3)] \\
&= P_{1\to 2}(R_1) E_{\alpha_1}[P_{2\to 3}(R_1, \alpha_1) E_{\alpha_2|R_1,\alpha_1}[E(Q|R_1, \alpha_1, \alpha_2)]] \\
&= \; \cdots \\
&= P_{1\to 2}(R_1) E_{\alpha_1}[P_{2\to 3}(R_1, \vec{\alpha}_1) E_{\alpha_2}[P_{3\to 4}(R_1, \vec{\alpha}_2) \; \cdots \\
&\qquad \cdots \; P_{\tilde{r}-1\to\tilde{r}}(R_1, \vec{\alpha}_{\tilde{r}-2}) E_{\alpha_{\tilde{r}-1}}(Q|R_1, \vec{\alpha}_{\tilde{r}-1})] \cdots] \\
&= E_{\vec{\alpha}_{\tilde{r}-1}}[P_{1\to 2}(R_1) P_{2\to 3}(R_1, \vec{\alpha}_1) \; \cdots \; P_{\tilde{r}-1\to\tilde{r}}(R_1, \vec{\alpha}_{\tilde{r}-2}) \\
&\qquad\qquad E(Q|R_1, \vec{\alpha}_{\tilde{r}-1})] \\
&\overset{(20)(21)}{=} E_{\vec{\alpha}_{\tilde{r}-1}} \left[\left(\prod_{u=1}^{\tilde{r}-1} P_{u\to u+1}(R_1, \vec{\alpha}_{u-1}) \right) E(H|R_1, \vec{\alpha}_{\tilde{r}-1}) \right] \\
&\overset{(109)}{=} E_{\vec{\alpha}_{\tilde{r}-1}}[P_{\text{DL}}(R_1, \vec{\alpha}_{\tilde{r}-2}) \, E(H|R_1, \vec{\alpha}_{\tilde{r}-1})] \\
&\overset{(23)}{=} E_{\vec{\alpha}_{\tilde{r}-1}}[P_{\text{DL}}(R_1, \vec{\alpha}_{\tilde{r}-2}) \, E(l\, A_{\tilde{r}}|R_1, \vec{\alpha}_{\tilde{r}-1})] \\
&\overset{\text{Remark 1}}{=} E_{\vec{\alpha}_{\tilde{r}-1}}[P_{\text{DL}}(R_1, \vec{\alpha}_{\tilde{r}-2}) \, l \, E(A_{\tilde{r}}|\vec{\alpha}_{\tilde{r}-1})] \\
&= E_{\vec{\alpha}_{\tilde{r}-1}}[G(R_1, \vec{\alpha}_{\tilde{r}-1})] \; ,
\end{aligned} \tag{114}$$

where

$$G(R_1, \vec{\alpha}_{\tilde{r}-1}) \; \triangleq \; l \; P_{\text{DL}}(R_1, \vec{\alpha}_{\tilde{r}-2}) \; E(A_{\tilde{r}}|\vec{\alpha}_{\tilde{r}-1}) \; . \tag{115}$$

Using (27) and (111), (115) yields

$$G(R_1, \vec{\alpha}_{\tilde{r}-1}) \; \approx \; l \; c \; (\lambda\, b_1\, R_1)^{\tilde{r}-1} \prod_{u=1}^{\tilde{r}-1} \frac{\tilde{n}_u}{b_u} \, (V_u\, \alpha_u)^{\tilde{r}-u} \; . \tag{116}$$

Unconditioning (116) on $\vec{\alpha}_{\tilde{r}-1}$, and given that the elements of $\vec{\alpha}_{\tilde{r}-1}$ are independent random variables approximately distributed according to (24) such that $E(\alpha_u^k) \approx 1/(k+1)$, (114) yields

$$E(Q|R_1) \approx \; l \; c \; (\lambda\, b_1\, R_1)^{\tilde{r}-1} \; \frac{1}{\tilde{r}\,!} \prod_{u=1}^{\tilde{r}-1} \frac{\tilde{n}_u}{b_u} \, V_u^{\tilde{r}-u} \; . \tag{117}$$

The expected amount of data lost, $E(Q)$, upon a first-device failure is obtained by unconditioning $E(Q|R_1)$ on R_1, that is,

$$E(Q) = E[E(Q|R_1)] \; . \tag{118}$$

Unconditioning (117) on R_1 using (10) and (34), (118) yields (47). ■

Appendix C
Approximate Derivation of $E(H)^{\text{sym}}$

Proof of Proposition 4.

First, we derive an approximation of the product

$$A \triangleq \prod_{j=1}^{m-l} \frac{m-j}{k-j} \; , \tag{119}$$

which appears in Equation (83). From (60), it follows that $m - l = hm$, as stated by (80). Substituting the preceding into (119), and using (61), yields

$$A = \prod_{j=1}^{hxk} \frac{x - \frac{j}{k}}{1 - \frac{j}{k}} \; , \tag{120}$$

or equivalently,

$$\log(A) = \sum_{j=1}^{hxk} \log\left(\frac{x - \frac{j}{k}}{1 - \frac{j}{k}} \right) \; , \tag{121}$$

To evaluate the preceding summation, we first establish the following lemmas.

Lemma 1: For small values of ϵ, that is, when ϵ approaches zero, and for any function $f(y)$, it holds that

$$\epsilon \sum_{j=1}^{\alpha/\epsilon} f(j\epsilon) \; \approx \; \int_{\frac{\epsilon}{2}}^{\alpha+\frac{\epsilon}{2}} f(y)\, dy \; , \qquad \forall\, \alpha \in \mathbb{R} \; . \tag{122}$$

Proof: The left-hand side of (122) is written as follows:

$$\epsilon \sum_{j=1}^{\alpha/\epsilon} f(j\epsilon) = \sum_{j=1}^{\alpha/\epsilon} f(j\epsilon)\, \epsilon \; . \tag{123}$$

For small small values of ϵ, the summation in the right-hand side of (123) represents the middle Riemann sum that approximates the definite integral of the $f(y)$ function in the interval $[\epsilon/2,\, \alpha + \epsilon/2]$, that is,

$$\sum_{j=1}^{\alpha/\epsilon} f(j\epsilon)\, \epsilon \; \approx \; \int_{\frac{\epsilon}{2}}^{\alpha+\frac{\epsilon}{2}} f(y)\, dy \; . \tag{124}$$

□

LEMMA 2: For any functions $f(y)$ and $F(y)$, such that $F(y) = \int f(y)\,dy$, or $F'(y) = f(y)$, and for $\alpha \in \mathbb{R}$ define

$$F^{(1)}(\alpha, z) \triangleq \int_{\frac{z}{2}}^{\alpha+\frac{z}{2}} f(y)\,dy = F\left(\alpha + \frac{z}{2}\right) - F\left(\frac{z}{2}\right). \tag{125}$$

Then it holds that

$$F_z^{(1)}(\alpha, z) = \frac{1}{2}\left[f\left(\alpha + \frac{z}{2}\right) - f\left(\frac{z}{2}\right)\right], \tag{126}$$

and

$$F_{zz}^{(1)}(\alpha, z) = \frac{1}{4}\left[f'\left(\alpha + \frac{z}{2}\right) - f'\left(\frac{z}{2}\right)\right]. \tag{127}$$

Proof: Immediate from the fact that for any $\alpha \in \mathbb{R}$ and function $f(y)$, it holds that $df(\alpha+z/2)/dz = f'(\alpha+z/2)/2$. □

Corollary 1: For $f(y) = \log(x - y)$ and for all $\alpha \in \mathbb{R}$, it holds that

$$F^{(1)}(x, \alpha, z) = \int_{\frac{z}{2}}^{\alpha+\frac{z}{2}} \log(x - y)\,dy = G(x, \alpha, z), \tag{128}$$

where

$$G(x, \alpha, z) \triangleq \log\left(\frac{(x - \frac{z}{2})^{x-\frac{z}{2}}}{(x - \alpha - \frac{z}{2})^{x-\alpha-\frac{z}{2}}}\right) - \alpha. \tag{129}$$

Also,

$$F_z^{(1)}(x, \alpha, z) = G_z(x, \alpha, z) = \frac{1}{2}\log\left(\frac{x - \alpha - \frac{z}{2}}{x - \frac{z}{2}}\right), \tag{130}$$

and

$$F_{zz}^{(1)}(x, \alpha, z) = G_{zz}(x, \alpha, z) = -\frac{\alpha}{4\,(x - \alpha - \frac{z}{2})(x - \frac{z}{2})}. \tag{131}$$

Proof: Equations (128) and (129) are derived from (125) by taking $f(y) = \log(x-y)$ and using the fact that $\int \log(y)\,dy = y\,[\log(y) - 1]$, which in turn implies that $F(y) = \int \log(x - y)\,dy = -(x - y)\,[\log(x - y) - 1]$. Equation (130) is directly obtained from (126), and (131) is obtained from (127) by using the fact that $f'(y) = -1/(x - y)$. □

Note that an approximation of $G(x, \alpha, z)$ for $z \approxeq 0$ can be obtained through its Maclaurin series as follows:

$$G(x, \alpha, z) \approx G(x, \alpha, 0) + G_z(x, \alpha, 0)\,z + \frac{G_{zz}(x, \alpha, 0)}{2}\,z^2, \tag{132}$$

which by virtue of (129), (130), and (131) yields

$$G(x, \alpha, z) \approx \log\left(\frac{x^x}{(x - \alpha)^{x-\alpha}}\right) - \alpha + \frac{1}{2}\log\left(\frac{x - \alpha}{x}\right)z - \frac{\alpha}{8(x - \alpha)x}\,z^2. \tag{133}$$

We now proceed with the evaluation of $\log(A)$. From (122) and (128), it follows that

$$\epsilon \sum_{j=1}^{\alpha/\epsilon} \log\left(\frac{x - j\epsilon}{1 - j\epsilon}\right) = G(x, \alpha, \epsilon) - G(1, \alpha, \epsilon). \tag{134}$$

From (121) and (129), and using (134) with $\epsilon = 1/k$ and $\alpha = hx$, we get

$$\log(A) \approx k\,F\left(x, \frac{1}{2k}\right), \tag{135}$$

where

$$F(x, y) \triangleq \log\left(\frac{(x - y)^{x-y}\,(1 - hx - y)^{1-hx-y}}{(1 - y)^{1-y}\,[(1 - h)x - y]^{(1-h)x-y}}\right). \tag{136}$$

An expression for $\log(A)$ for large values of k, m, l, and $m - l$ can be obtained from (121) and (134) by using approximation (133) with $\epsilon = 1/k$ and $\alpha = hx$ as follows:

$$\log(A) \approx k\,\log\left(\frac{x^x\,(1 - hx)^{1-hx}}{[(1 - h)x]^{(1-h)x}}\right) + \log\left(\sqrt{\frac{1 - h}{1 - hx}}\right) - \frac{1}{k}\,\frac{h(1 - x)[1 + (1 - h)x]}{8(1 - h)(1 - hx)x}. \tag{137}$$

Equation (58) is a direct consequence of (83) and also of (79), (80), (119), and (137), where the third term of the summation in (137) is ignored for large k. ■

APPENDIX D
APPROXIMATE DERIVATION OF MTTDL$^{\text{sym}}$

Proof of Proposition 5.

Using (79) and (80), (81) can be written as follows:

$$\frac{n\,\lambda\,\text{MTTDL}^{\text{sym}}}{k} \approx \frac{1}{k}\left[\frac{b}{[(1 - h)xk + 1]\,\lambda\,c}\right]^{hxk} (hxk)!\;\frac{[E(X)]^{hxk}}{E(X^{hxk})}\prod_{j=1}^{m-l}\left(\frac{k - j}{m - j}\right)^{m-l-j}. \tag{138}$$

From (138), and using Stirling's approximation (139) for large values of k, with k replaced by hxk, that is

$$(hxk)! \approx \sqrt{2\pi hxk}\,\left(\frac{hxk}{e}\right)^{hxk}, \tag{139}$$

it follows that

$$\log\left(\frac{n\,\lambda\,\text{MTTDL}_{\text{approx}}^{\text{sym}}}{k}\right) \approx \log\left(\sqrt{\frac{2\pi hx}{k}}\right) + hxk\,\log\left(\frac{hxk\,b}{e\,[(1 - h)xk + 1]\,\lambda\,c}\right) + \log\left(\frac{[E(X)]^{hxk}}{E(X^{hxk})}\right) + \log(B), \tag{140}$$

where B is the product

$$B \triangleq \prod_{j=1}^{m-l}\left(\frac{k - j}{m - j}\right)^{m-l-j}. \tag{141}$$

We now proceed to derive an approximation of the product B. By virtue of (61), (80), and (119), the product B can be

written as follows:

$$B = \frac{\prod_{j=1}^{m-l} \left(\frac{m-j}{k-j}\right)^j}{\prod_{j=1}^{m-l} \left(\frac{m-j}{k-j}\right)^{m-l}} = \frac{C}{A^{m-l}} = \frac{C}{A^{hxk}} \ , \qquad (142)$$

where

$$C \triangleq \prod_{j=1}^{m-l} \left(\frac{m-j}{k-j}\right)^j = \prod_{j=1}^{hxk} \left(\frac{x-\frac{j}{k}}{1-\frac{j}{k}}\right)^j \ . \qquad (143)$$

From (142) and (143), it follows that

$$\log(B) = \log(C) - hxk \, \log(A) \qquad (144)$$

and

$$\log(C) = \sum_{j=1}^{hxk} j \, \log\left(\frac{x-\frac{j}{k}}{1-\frac{j}{k}}\right) \ . \qquad (145)$$

To evaluate the preceding summation, we first establish the following corollary from Lemma 2.

Corollary 2: For $f(y) = y \log(x-y)$ and for all $\alpha \in \mathbb{R}$, it holds that

$$F^{(1)}(x,\alpha,z) = \int_{\frac{z}{2}}^{\alpha+\frac{z}{2}} y \log(x-y) \, dy = R(x,\alpha,z) \ , \qquad (146)$$

where

$$R(x,\alpha,z) \triangleq \frac{1}{2} \log\left(\frac{(x-\frac{z}{2})^{x^2-(\frac{z}{2})^2}}{(x-\alpha-\frac{z}{2})^{x^2-(\alpha+\frac{z}{2})^2}}\right) - \frac{\alpha(2x+\alpha+z)}{4} . \qquad (147)$$

Also,

$$F_z^{(1)}(x,\alpha,z) = R_z(x,\alpha,z) = \frac{1}{2} \log\left(\frac{(x-\alpha-\frac{z}{2})^{\alpha+\frac{z}{2}}}{(x-\frac{z}{2})^{\frac{z}{2}}}\right) \qquad (148)$$

and

$$F_{zz}^{(1)}(x,\alpha,z) = R_{zz}(x,\alpha,z) = \frac{1}{4}\left[\log\left(\frac{x-\alpha-\frac{z}{2}}{x-\frac{z}{2}}\right) - \frac{\alpha x}{(x-\alpha-\frac{z}{2})(x-\frac{z}{2})}\right] \ . \qquad (149)$$

Proof: Equations (146) and (147) are derived from (125) by taking $f(y) = y\log(x-y)$ and using the fact that $\int y\log(y)\,dy = y^2\,(2\log(y)-1)/4$, which in turn implies that $F(y) = \int y\log(x-y)\,dy = (x-y)\,[3x+y-2(x+y)\log(x-y)]/4$. Equation (148) is directly obtained from (126), and (149) is obtained from (127) by using the fact that $f'(y) = \log(x-y) - y/(x-y)$. □

Note that an approximation of $R(x,\alpha,z)$ for $z \cong 0$ can be obtained through its Maclaurin series as follows:

$$R(x,\alpha,z) \approx R(x,\alpha,0) + R_z(x,\alpha,0)\,z + \frac{R_{zz}(x,\alpha,0)}{2} z^2 \ , \qquad (150)$$

which by virtue of (147), (148), and (149) yields

$$R(x,\alpha,z) \approx \frac{1}{2}\log\left(\frac{x^{x^2}}{(x-\alpha)^{x^2-\alpha^2}}\right) - \frac{\alpha(2x+\alpha)}{4} + \frac{\alpha}{2}\log(x-\alpha)\,z + \frac{1}{8}\left[\log\left(\frac{x-\alpha}{x}\right) - \frac{\alpha}{(x-\alpha)x}\right] z^2 \ . \qquad (151)$$

We now proceed with the evaluation of $\log(C)$. From (122) and (146), it follows that

$$\epsilon \sum_{j=1}^{\alpha/\epsilon} j \, \log\left(\frac{x-j\epsilon}{1-j\epsilon}\right) = R(x,\alpha,\epsilon) - R(1,\alpha,\epsilon) \ . \qquad (152)$$

From (121) and (147), and using (152) with $\epsilon = 1/k$ and $\alpha = hx$, we get

$$\log(C) \approx k^2 \, \frac{1}{2} \left[hx(1-x) + S\left(x, \frac{1}{2k}\right)\right] \ , \qquad (153)$$

where

$$S(x,y) \triangleq \log\left(\frac{(x-y)^{x^2-y^2}(1-hx-y)^{1-(hx+y)^2}}{(1-y)^{1-y^2}\,[(1-h)x-y]^{x^2-(hx+y)^2}}\right) . \qquad (154)$$

An expression for $\log(C)$ for large values of k, m, l, and $m-l$ can be obtained from (145) and (152) by using approximation (151) with $\epsilon = 1/k$ and $\alpha = hx$ as follows:

$$\log(C) \approx \frac{k^2}{2}\left[hx(1-x) + \log\left(\frac{x^{x^2}\,(1-hx)^{1-(hx)^2}}{[(1-h)x]^{(1-h^2)x^2}}\right)\right] + \frac{k}{2}\,hx\log\left(\frac{(1-h)x}{1-hx}\right) + \frac{1}{8}\log\left(\frac{1-h}{1-hx}\right) - \frac{h(1-x)}{8(1-h)(1-hx)} \ . \qquad (155)$$

Substituting (137) and (155) into (144) yields

$$\log(B) \approx \frac{k^2}{2}\left[hx(1-x) - \log\left(\frac{\left[x^{h^2}\,(1-h)^{(1-h)^2}\right]^{x^2}}{(1-hx)^{(1-hx)^2}}\right)\right] + k\,hx\log(\sqrt{x}) - \frac{1}{8}\left[h(1-x) - \log\left(\frac{1-h}{1-hx}\right)\right] \ . \qquad (156)$$

Equation (62) is a direct consequence of (140) and (156). ■

Appendix E
Approximate Derivation of EAFDL$^{\text{sym}}$

Proof of Proposition 6.

From (15), it follows that

$$\text{EAFDL}/\lambda = \frac{E(H)/c}{\lambda\,\text{MTTDL}\,\cdot\,U\,\cdot c} \ . \qquad (157)$$

Substituting (2) into (157), and using (60), yields

$$\text{EAFDL}/\lambda = \frac{E(H)/c}{\lambda\,\text{MTTDL}\,\cdot\,(1-h)\,n} \qquad (158)$$

or

$$\log(\text{EAFDL}/\lambda) = \log(E(H)/c) - \log(\lambda\,\text{MTTDL}) - \log((1-h)\,n) \ . \qquad (159)$$

Substituting (58) and (62) into (159), after some manipulations yields (64). ■

APPENDIX F
OPTIMAL CODEWORD LENGTHS FOR MAXIMIZING MTTDL$^{\text{declus}}$ AND MINIMIZING EAFDL$^{\text{declus}}$ FOR A LARGE NUMBER OF STORAGE DEVICES, n

Proof of Proposition 11.

We first consider the optimal codeword lengths for maximizing MTTDL$^{\text{declus}}$. From (103), it holds that

$$r^*_{\text{MTTDL}}(n) = \frac{m^*_{\text{MTTDL}}(n)}{n} = \frac{\arg \max\limits_{1\le m\le n} \text{MTTDL}^{\text{declus}}}{n} . \tag{160}$$

Using (73), the preceding can be written as follows:

$$r^*_{\text{MTTDL}}(n) = \arg \max_{\frac{1}{n}\le x\le 1} \text{MTTDL}^{\text{declus}} \tag{161}$$

or

$$r^*_{\text{MTTDL}}(n) = \arg \max_{\frac{1}{n}\le x\le 1} \log(\lambda\, \text{MTTDL}^{\text{declus}}) \tag{162}$$

or, equivalently,

$$r^*_{\text{MTTDL}}(n) = \arg \max_{\frac{1}{n}\le x\le 1} \left(\frac{2\,\log(\lambda\, \text{MTTDL}^{\text{declus}})}{n^2} \right) . \tag{163}$$

By letting n approach the infinity, we get

$$\begin{aligned} r^*_{\infty} &= \lim_{n\to\infty} r^*_{\text{MTTDL}}(n) \\ &= \lim_{n\to\infty} \arg \max_{\frac{1}{n}\le x\le 1} \left(\frac{2\,\log(\lambda\, \text{MTTDL}^{\text{declus}})}{n^2} \right) \\ &= \arg \max_{0<x\le 1} \lim_{n\to\infty} \left(\frac{2\,\log(\lambda\, \text{MTTDL}^{\text{declus}})}{n^2} \right) . \end{aligned} \tag{164}$$

Using the approximation obtained in (85), (164) yields

$$r^*_{\infty} = \arg \max_{0<x\le 1} W(h,x) , \tag{165}$$

provided that for the last term of the summation in (85) it holds that

$$\lim_{n\to\infty} \frac{\log\left(\dfrac{[E(X)]^{hxn}}{E(X^{hxn})} \right)}{n^2} = 0 . \tag{166}$$

Remark 13: It turns out that (166) holds for the cases of deterministic and exponential rebuild time distributions owing to the following lemmas.

LEMMA 3: In the case of deterministic rebuild times, it holds that

$$\log\left(\frac{[E(X)]^{hxn}}{E(X^{hxn})} \right) = 0 . \tag{167}$$

Proof: Equation (167) follows from the fact that in the case of deterministic rebuild times it holds that $E(X^{hxn}) = [E(X)]^{hxn}$. □

LEMMA 4: In the case of exponential rebuild times, it holds that

$$\frac{\log\left(\dfrac{[E(X)]^{hxn}}{E(X^{hxn})} \right)}{n^2} \approx hx\, \frac{\log\left(\dfrac{hxn}{e} \right)}{n} + \frac{\log(2\pi hxn)}{2\,n^2} . \tag{168}$$

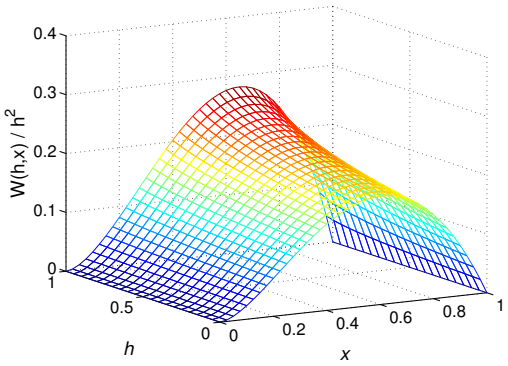


Figure 26. $W(h,x)/h^2$ for h and $x \in [0,1]$.

Proof: In the case of exponential rebuild times, it holds that $E(X^{hxn}) = (hxn)!\,[E(X)]^{hxn}$, which for large n and by virtue of (139), yields

$$\log\left(\frac{[E(X)]^{hxn}}{E(X^{hxn})} \right) \approx \log\left(\sqrt{2\pi hxn}\, \left(\frac{hxn}{e} \right)^{hxn} \right) . \tag{169}$$

Equation (168) follows directly from (169). □

From (63), it follows that $W(h,x)$ or, equivalently, $W(h,x)/h^2$ are non-negative in $x \in [0,1]$, with $W(h,0) = W(h,1) = 0$, as shown in Figure 26. Consequently, (165) implies that r^*_{∞} satisfies the following equation:

$$W_x(h, r^*_{\infty}) = \left. \frac{dW(h,x)}{dx} \right|_{x=r^*_{\infty}} = 0 . \tag{170}$$

The derivative of $W(h,x)$ with respect to x can be obtained using the following lemma.

LEMMA 5: For $w(y)$ defined as follows:

$$w(y) = \log\left(f(y)^{g(y)} \right) = \log\left(f^g \right) , \tag{171}$$

it holds that

$$w'(y) = w' = g' \log(f) + g f'/f . \tag{172}$$

Corollary 3: For $v(y)$ defined as follows:

$$v(y) = \log\left(f(y)^{f(y)} \right) = \log\left(f^f \right) , \tag{173}$$

it holds that

$$v'(y) = v' = f' \left(\log(f) + 1 \right) . \tag{174}$$

From (63), it follows that the derivative of $W(h,x)$ with respect to x can be obtained by successively applying (172), which yields

$$W_x(h,x) = -2\, Q(h,x) , \tag{175}$$

where $Q(h,x)$ is given by (105). Thus, r^*_{∞} is obtained as the unique root of the equation $Q(h,x) = 0$, with respect to x, in the interval $(0,1]$, that is,

$$Q(h, r^*_{\infty}) = 0 , \quad \text{with} \quad r^*_{\infty} \in (0,1] . \tag{176}$$

The values of r^*_{∞} as a function of h are shown in Figure 27.

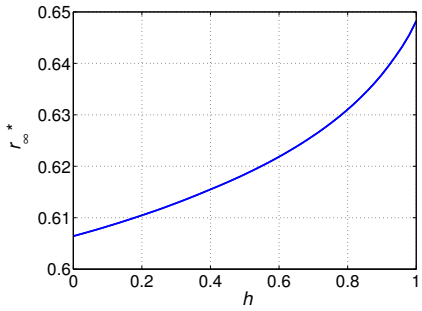


Figure 27. r^*_∞ vs. h for MTTDL and EAFDL.

Remark 14: For $h = 0$, it holds that $Q(0, x) = 0$. To find the root when $h \to 0$, we consider finding the root of the equivalent equation $Q(h, x)/h^2 = 0$. Using L'Hôpital's rule, after some manipulations, we obtain

$$\lim_{h\to 0} \frac{Q(h,x)}{h^2} = x\left(\log(x) + \frac{1}{2}\right) = x\log(\sqrt{e}\,x)\,. \quad (177)$$

Combining (176) and (177) yields

$$r^*_\infty \log(\sqrt{e}\,r^*_\infty) = 0\,, \quad \text{with } r^*_\infty \in (0,1]$$
$$\text{or } r^*_\infty = \frac{1}{\sqrt{e}} = 0.606\,. \quad (178)$$

For $h = 1$, r^*_∞ is obtained as the unique root in $(0, 1]$ of the equation

$$Q(1,x) = x + \log\left(x^x(1-x)^{1-x}\right) = 0\,, \quad (179)$$

which yields $r^*_\infty = 0.648$.

We now proceed to derive the optimal codeword lengths for maximizing the EAFDL$^{\text{declus}}$ for large values of n, m, l, and $m - l$. Analogously to (164), it holds that

$$r^*_\infty = \lim_{n\to\infty} r^*_{\text{EAFDL}}(n)$$
$$= \arg\min_{0<x\le 1} \lim_{n\to\infty} \left(\frac{2\,\log(\text{EAFDL}^{\text{declus}}/\lambda)}{n^2}\right)\,. \quad (180)$$

Using the approximation obtained in (86), (180) yields

$$r^*_\infty = \arg\max_{0<x\le 1} W(h,x)\,, \quad (181)$$

provided that for the last term of the summation in (86) the condition given by (166) holds. Given that (181) is the same as (165), we deduce that the r^*_∞ values for EAFDL are the same as those for MTTDL. ■

Appendix G
Optimal Codeword Length for Minimizing $E(H)^{\text{declus}}$ for Large n

Proof of Proposition 12.

From (103), it holds that

$$r^*(n) = \frac{m^*(n)}{n} = \frac{\arg\min_{1\le m\le n} E(H)^{\text{declus}}}{n}\,. \quad (182)$$

Using (73), the preceding can be written as follows:

$$r^*(n) = \arg\min_{\frac{1}{n}\le x\le 1} E(H)^{\text{declus}} \quad (183)$$

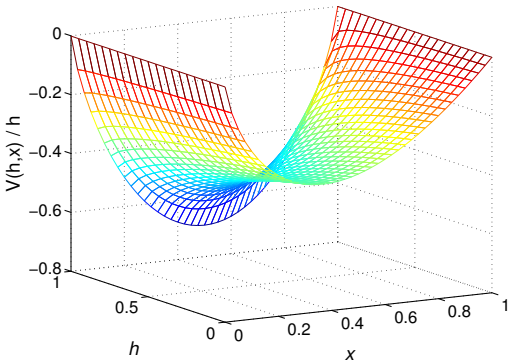


Figure 28. $V(h, x)/h$ for h and $x \in [0, 1]$.

or

$$r^*(n) = \arg\min_{\frac{1}{n}\le x\le 1} \log(E(H)^{\text{declus}}/c) \quad (184)$$

or, equivalently,

$$r^*(n) = \arg\min_{\frac{1}{n}\le x\le 1} \left(\frac{\log(E(H)^{\text{declus}}/c)}{n}\right)\,. \quad (185)$$

By letting n approach the infinity we get

$$r^*_\infty = \lim_{n\to\infty} r^*(n) = \lim_{n\to\infty} \arg\min_{\frac{1}{n}\le x\le 1} \left(\frac{\log(E(H)^{\text{declus}}/c)}{n}\right)$$
$$= \arg\min_{0<x\le 1} \lim_{n\to\infty} \left(\frac{\log(E(H)^{\text{declus}}/c)}{n}\right)\,. \quad (186)$$

Using the approximation obtained in (87), (186) yields

$$r^*_\infty = \arg\min_{0<x\le 1} V(h,x)\,. \quad (187)$$

From (59), it follows that $V(h, x)$ or, equivalently, $V(h, x)/h$ are convex in $x \in [0, 1]$, with $V(h, 0) = V(h, 1) = 0$, as shown in Figure 28. Consequently, (187) implies that r^*_∞ satisfies the following equation:

$$V_x(h, r^*_\infty) = \left.\frac{dV(h,x)}{dx}\right|_{x=r^*_\infty} = 0\,. \quad (188)$$

From (59), it follows that the derivative of $V(h, x)$ with respect to x can be obtained using Corollary 3. By successively applying (174), with $f(x)$ being equal to x, $1 - hx$, and $(1 - h)x$, yields

$$V_x(h,x) = \log\left(\frac{x\,(1-hx)^{-h}}{[(1-h)x]^{1-h}}\right)\,. \quad (189)$$

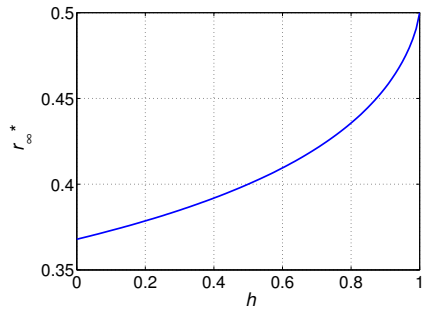


Figure 29. r^*_∞ vs. h for $E(H)$.

From (188) and (189), we deduce that r^*_∞ satisfies the following equation:

$$\log\left(\frac{r^*_\infty\,(1-h\,r^*_\infty)^{-h}}{[(1-h)r^*_\infty]^{1-h}}\right) = 0\ . \qquad (190)$$

Solving (190) for r^*_∞ yields (107), which is shown in Figure 29. ■

REFERENCES

[1] I. Iliadis and V. Venkatesan, "Reliability assessment of erasure coded systems," in Proceedings of the 10th International Conference on Communication Theory, Reliability, and Quality of Service (CTRQ), Apr. 2017, pp. 41–50.

[2] D. A. Patterson, G. Gibson, and R. H. Katz, "A case for redundant arrays of inexpensive disks (RAID)," in Proceedings of the ACM SIGMOD International Conference on Management of Data, Jun. 1988, pp. 109–116.

[3] P. M. Chen, E. K. Lee, G. A. Gibson, R. H. Katz, and D. A. Patterson, "RAID: High-performance, reliable secondary storage," ACM Comput. Surv., vol. 26, no. 2, Jun. 1994, pp. 145–185.

[4] M. Malhotra and K. S. Trivedi, "Reliability analysis of redundant arrays of inexpensive disks," J. Parallel Distrib. Comput., vol. 17, Jan. 1993, pp. 146–151.

[5] W. A. Burkhard and J. Menon, "Disk array storage system reliability," in Proceedings of the 23rd International Symposium on Fault-Tolerant Computing, Jun. 1993, pp. 432–441.

[6] K. S. Trivedi, Probabilistic and Statistics with Reliability, Queueing and Computer Science Applications, 2nd ed. New York: Wiley, 2002.

[7] Q. Xin, E. L. Miller, T. J. E. Schwarz, D. D. E. Long, S. A. Brandt, and W. Litwin, "Reliability mechanisms for very large storage systems," in Proceedings of the 20th IEEE/11th NASA Goddard Conference on Mass Storage Systems and Technologies (MSST), Apr. 2003, pp. 146–156.

[8] 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.

[9] S. Ramabhadran and J. Pasquale, "Analysis of long-running replicated systems," in Proc. 25th IEEE International Conference on Computer Communications (INFOCOM), Apr. 2006, pp. 1–9.

[10] B. Eckart, X. Chen, X. He, and S. L. Scott, "Failure prediction models for proactive fault tolerance within storage systems," in Proceedings of the 16th Annual IEEE International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS), Sep. 2008, pp. 1–8.

[11] K. Rao, J. L. Hafner, and R. A. Golding, "Reliability for networked storage nodes," IEEE Trans. Dependable Secure Comput., vol. 8, no. 3, May 2011, pp. 404–418.

[12] J.-F. Pâris, T. J. E. Schwarz, A. Amer, and D. D. E. Long, "Highly reliable two-dimensional RAID arrays for archival storage," in Proceedings of the 31st IEEE International Performance Computing and Communications Conference (IPCCC), Dec. 2012, pp. 324–331.

[13] I. Iliadis and V. Venkatesan, "An efficient method for reliability evaluation of data storage systems," in Proceedings of the 8th International Conference on Communication Theory, Reliability, and Quality of Service (CTRQ), Apr. 2015, pp. 6–12.

[14] ——, "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.

[15] V. Venkatesan and I. Iliadis, "A general reliability model for data storage systems," in Proceedings of the 9th International Conference on Quantitative Evaluation of Systems (QEST), Sep. 2012, pp. 209–219.

[16] 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.

[17] A. Thomasian and M. Blaum, "Higher reliability redundant disk arrays: Organization, operation, and coding," ACM Trans. Storage, vol. 5, no. 3, Nov. 2009, pp. 1–59.

[18] K. M. Greenan, J. S. Plank, and J. J. Wylie, "Mean time to meaningless: MTTDL, Markov models, and storage system reliability," in Proceedings of the USENIX Workshop on Hot Topics in Storage and File Systems (HotStorage), Jun. 2010, pp. 1–5.

[19] 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.

[20] I. Iliadis and V. Venkatesan, "Rebuttal to 'Beyond MTTDL: A closed-form RAID-6 reliability equation'," ACM Trans. Storage, vol. 11, no. 2, Mar. 2015, pp. 1–10.

[21] "Amazon Simple Storage Service." [Online]. Available: http://aws.amazon.com/s3/ [retrieved: November 2017]

[22] D. Borthakur et al., "Apache Hadoop goes realtime at Facebook," in Proceedings of the ACM SIGMOD International Conference on Management of Data, Jun. 2011, pp. 1071–1080.

[23] R. J. Chansler, "Data availability and durability with the Hadoop Distributed File System," ;login: The USENIX Association Newsletter, vol. 37, no. 1, 2013, pp. 16–22.

[24] K. Shvachko, H. Kuang, S. Radia, and R. Chansler, "The Hadoop Distributed File System," in Proceedings of the 26th IEEE Symposium on Mass Storage Systems and Technologies (MSST), May 2010, pp. 1–10.

[25] 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.

[26] C. Huang et al., "Erasure coding in Windows Azure Storage," in Proceedings of the USENIX Annual Technical Conference (ATC), Jun. 2012, pp. 15–26.

[27] "IBM Cloud Object Storage." [Online]. Available: www.ibm.com/cloud-computing/products/storage/object-storage/how-it-works/ [retrieved: November 2017]

[28] V. Venkatesan and I. Iliadis, "Effect of codeword placement on the reliability of erasure coded data storage systems," in Proceedings of the 10th International Conference on Quantitative Evaluation of Systems (QEST), Sep. 2013, pp. 241–257.

[29] H. Weatherspoon and J. Kubiatowicz, "Erasure coding vs. replication: A quantitative comparison," in Proceedings of the 1st International Workshop on Peer-to-Peer Systems (IPTPS), Mar. 2002, pp. 328–338.

[30] R. Rodrigues and B. Liskov, "High availability in DHTs: Erasure coding vs. replication," in Proceedings of the 4th International Workshop on Peer-to-Peer Systems (IPTPS), Feb. 2005, pp. 226–239.

[31] J. S. Plank and C. Huang, "Tutorial: Erasure coding for storage applications," Slides presented at 11th Usenix Conference on File and Storage Technologies (FAST'13), San Jose, CA, Feb. 2013.

[32] V. Venkatesan, I. Iliadis, C. Fragouli, and R. Urbanke, "Reliability of clustered vs. declustered replica placement in data storage systems," in Proceedings of the 19th Annual IEEE/ACM International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS), Jul. 2011, pp. 307–317.

[33] V. Venkatesan, I. Iliadis, and R. Haas, "Reliability of data storage systems under network rebuild bandwidth constraints," in Proceedings of the 20th Annual IEEE International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS), Aug. 2012, pp. 189–197.

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

[35] V. Venkatesan and I. Iliadis, "Effect of codeword placement on the reliability of erasure coded data storage systems," IBM Research Report, RZ 3827, Aug. 2012.