Effect of Error-Correction Coding Schemes on Maximum Throughput of Automated Tape Systems

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

***Abstract*—Tape is a highly scalable and reliable media, which enables long-term access of stored data. Tape's low energy consumption combined with the low cost per terabyte make it an appealing option for storing infrequently accessed data. Power and operational failures may damage tapes and lead to data loss. To protect stored data against loss, erasure-coding redundancy schemes are employed. Performance is affected by the size and operational characteristics of the tape libraries, the mount and unmount policies employed, the request servicing policy adopted, the erasure coding configurations implemented, and the characteristics of the workload considered. In this article, we develop a theoretical model that takes into account the principal operational aspects of a tape library system and derives its maximum throughput. It is demonstrated that employing erasure coding may adversely affect the maximum throughput. It is also established that the maximum throughput improves when requests are served according to schemes that effectively reduce seek times. The model provides useful insights into the effect of various system configurations and yields results that enable a better understanding of the design tradeoffs between reliability and system processing capability reflected by the maximum throughput.**

Keywords–Performance analysis; Reliability; MTTDL; EAFEDL; MDS codes; Unrecoverable or latent sector errors; stochastic modeling.

I. INTRODUCTION

Modern tape systems are well suited for storing infrequently accessed data in the context of cold and active archives, backup and disaster recovery [1]. Moreover, tape is a highly scalable and reliable media, which enables long-term access to stored data [2]. Tape's key advantage over hard-disk drives (HDDs) and flash is its low cost per gigabyte and substantial energy savings. State-of-the-art enterprise tape drives operate with a native cartridge capacity of 50 TB and data rates of 400 MB/s. Tape storage also offers high data security owing to a built-in physical air gap, which improves cyber resilience by isolating tape media from direct access. Security can be further enhanced by exporting cartridges to an off-site vault.

Today's storage systems and most cloud offerings employ redundancy and recovery schemes to protect stored data against device failures and media errors. In particular, high data reliability is achieved by employing efficient erasure-coding redundancy schemes [3-6]. The effectiveness of these schemes has been evaluated based on the Mean Time to Data Loss (MTTDL) and the Expected Annual Fraction of Data Loss (EAFDL) metric. This metric was recently complemented by the Expected Annual Fraction of Effective Data Loss (EAFEDL) metric, which assesses data losses at an *entity*, say file, object, or block level, whereas the EAFDL metric assesses data losses at a lower data processing unit level [7][8]. The MTTDL and EAFEDL metrics provide a useful profile of the frequency and magnitude of data losses. These metrics were recently used to assess the reliability of automated tape library systems [9].

Most traditional tape architectures achieve additional data protection through redundancy, that is, additional copies of a tape. More recently, a variety of tape software solutions have provided an option for cartridge level error-correction coding, an approach that is often referred to as RAIL (Redundant Array of Independent Libraries) [10] or RAIT (Redundant Array of independent Tapes/tape drives). Examples include HPSS (High Performance Storage System) [11], which supports many erasure-code rate options with data stripe widths up to 15 and parity stripe widths up to 7, and PoINT Archival Gateway [12] with erasure-code rate options of 2/3, 2/4 and 3/4. Most currently available solutions with error correction for tape use the MDS (Maximum Distance Separable) coding as analyzed in the present study. More complex schemes such as locally repairable codes [13] are beyond the scope of this work. Also, beyond the scope of this work is a detailed analysis of the tradeoff between error-correction coding schemes and power consumption. However, we can estimate the impact of these schemes by assuming that the power consumption of a given configuration is dominated by the number of tape drives used and then simply compare various configurations based on their corresponding number of tape drives.

The study presented in [9] suggests that power and operational failures are the main events that may damage tapes and lead to data loss. To protect stored data against loss and achieve high data reliability, an erasure-coding redundancy scheme is employed. A theoretical analytical model that considers the principal aspects of tape library operation and assesses the effect of tape failures on system reliability was presented in [9]. This model also captures the effect of latent errors, that is, uncorrectable errors that have not been detected.

In this article, we consider the operation of an original unprotected tape library system and derive the corresponding maximum entity throughput. Subsequently, we consider the employment of an erasure-coding redundancy scheme, as presented in [9], and study the effect of this scheme on the maximum entity throughput of the protected tape library system. A theoretical model that includes all the relevant parameters is developed. Requests submitted to tapes are queued and subsequently served according to a policy. In this work, we consider a spectrum of such scheduling policies for accessing data within a cartridge including the First-Come-First-Served (FCFS) and the Recommended Access Order (RAO) policy,

which schedules the requests to be served in an order that results in reduced seek times [14]. Therefore, these policies affect seek times that in turn affect the maximum throughput of the system. Closed-form expressions are derived for the maximum entity throughput by considering random and sequential workloads. We theoretically establish that, for random workloads, employing the RAO policy results in an increase of the maximum throughput. The results obtained demonstrate that the employment of erasure coding schemes with increased capability results in an improved system reliability, but does not necessarily lead to higher maximum throughputs.

The remainder of the article is organized as follows. Section II describes the storage system model and the corresponding parameters considered. Section III describes the operation of an original unprotected tape library system as well as that of a protected one that employs an erasure-coding redundancy scheme. In Section IV, the maximum throughputs of both the original unprotected and protected tape library systems are derived analytically as a function of the relevant system parameters and for a spectrum of scheduling policies including the FCFS and RAO ones. Section V presents numerical results demonstrating the effect of the erasure-coding capability and of the policy employed on the maximum throughput. Finally, we conclude in Section VI.

II. Tape Library System Modeling

A storage system is comprised of tape libraries with each tape library containing d tape drives, a robot arms, and comprising c tape cartridges, where each cartridge stores an amount C_t of data such that the total storage capacity of a tape library is $c\,C_t$.

The smallest accessed unit of a tape is a *data set* in Linear Tape-Open (LTO is the trademark of HP, IBM, and Quantum in the Unites States and other countries) tape systems (IBM is a registered trademark of International Business Machines Corporation, registered in many jurisdictions worldwide) [15]. A data set currently has a size of about 5 MB of user data or more. In particular, for the LTO-8 tape technology, a data set has a size of 6.119424 MB of encoded data of which 5.096448 MB are user data [16]. Erasure-coding redundancy schemes are implemented by treating the units that contain user data as symbols and complementing them with parity symbols (units) to form codewords.

User data are stored in *entities* of E_s different sizes, $e_{s,1}, e_{s,2}, \cdots, e_{s,E_s}$. Without loss of generality, we assume that $e_{s,1} < e_{s,2} < \cdots < e_{s,E_s}$. The corresponding probability density function $\{v_j\}$ of a typical entity size e_s is

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

Successive entity sizes are assumed to be independent and identically distributed (i.i.d) according to the distribution given in (1). The first moment of a random variable X is denoted by $\overline{X}$. Thus, $\overline{e_s}$ denotes the first moment of the entity size e_s given by

$$\overline{e_s} = \sum_{j=1}^{E_s} e_{s,j}\, v_j \ . \tag{2}$$

Depending on whether redundancy is introduced and a recovery scheme is employed to protect a system against data loss, two cases are considered (Sections II-A and II-B, respectively).

TABLE I. Notation of system parameters

Parameter	Definition
d	number of tape drives in a tape library
a	number of robot arms (accessors) in a tape library
C_t	amount of data stored on a tape cartridge
s	data set (symbol) size
e_s	entity size
L	number of tape libraries in original unprotected storage system
L_r	number of tape libraries in storage system with redundancy
c	number of tape cartridges in a tape library
l	number of user-data symbols per codeword ($l \geq 1$)
m	total number of symbols per codeword ($m > l$)
(m, l)	MDS-code structure
b_w	bandwidth (data rate)
t_L	load time
t_U	unload time
t_R	rewind time
$s_{\max}$	maximum seek time of a request at the end of tape
s_I	seek time of initial requests
s_N	seek time of non-initial requests
$s_{\text{s,N}}$	seek time of non-initial requests at saturation
λ_e	arrival rate of entity requests
s_{eff}	storage efficiency of redundancy scheme ($s_{\text{eff}} = l/m$)
U	amount of user data stored in system ($U = L\ c\ C_t$)
U_r	amount of user data stored in r-system ($U_r = s_{\text{eff}}\ L_r\ c\ C_t$)
N_E	number of entities in system ($N_E = (L\ c\ C_t)/(\overline{e_s})$)
$N_{\text{E,r}}$	number of entities in r-system ($N_{\text{E,r}} = (l\ L_r\ c\ C_t)/(m\,\overline{e_s})$)
$N_{\text{E,rc}}$	number of entities stored in a cluster ($N_{\text{E,rc}} = l\ C_t/\overline{e_s}$)
n	number of cartridges per tape drive ($n = c/d$)
k	maximum number of entities processed in system ($k = L\,d$)
k_r	number of tape drive groups in r-system ($k_r = \lfloor L_r\,d/l \rfloor$)
g_r	number of arrays/clusters in r-system ($g_r = \lfloor L_r\,c/m \rfloor$)
$\tilde{r}$	MDS-code distance: minimum number of codeword symbols lost that lead to permanent data loss ($\tilde{r} = m - l + 1$ and $2 \leq \tilde{r} \leq m$)
C	number of symbols stored in a device ($C = c/s$)
s_s	shard size ($s_s = e_s/l$)
M	mount time ($M = R + t_L$)
U	unmount time ($U = t_R + t_U + R$)
	transfer time of an arbitrary entity

A. No Data Recovery and Protection

An original unprotected storage system is comprised of L tape libraries such that the total storage capacity and the amount U of user data stored in the system is

$$U \ = \ L\,c\,C_t \ . \tag{3}$$

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

Therefore, the number N_E of entities in the system is

$$N_E \ \approx \ \frac{U}{\overline{e_s}} \ \overset{(3)}{=} \ \frac{L\,c\,C_t}{\overline{e_s}} \ . \tag{4}$$

Also, the number $N_{\text{E,c}}$ of entities in a tape is

$$N_{\text{E,c}} \ \approx \ \frac{C_t}{\overline{e_s}} \ . \tag{5}$$

To access an entity, the corresponding tape is mounted to a free drive. As the system comprises $L\,d$ tape drives, at any given time, there can be at most k entities processed, where

$$k \ = \ L\,d \ , \tag{6}$$

with the corresponding k tapes mounted to the k drives.

B. Data Recovery and Protection

To protect stored data against loss, redundancy schemes are employed. For reliability purposes, we consider Maximum Distance Separable (MDS) erasure codes (m, l), which are commonly used for both HDD and tape, that map l user-data symbols to codewords of m symbols. They have the property that any subset containing l of the m codeword symbols can be used to reconstruct (recover) a codeword. The corresponding storage efficiency s_{eff} is then given by

$$s_{\text{eff}} = l/m \, . \tag{7}$$

Erasure coding across tapes within a tape library is exposed to failure mechanisms such as robot failures and network failures. As erasure coding across multiple libraries provides redundancy against such failure mechanisms, we proceed by considering multiple tape libraries in our analysis. Consequently, the number L_r of tape libraries required to store the user and additional parity data is larger than L and is determined by

$$L_r = L \, / s_{\text{eff}} \overset{(7)}{=} m \, L \, / l \, . \tag{8}$$

The amount U_r of user data stored in the system is

$$U_r = U \overset{(3)}{=} L \, c \, C_t \overset{(8)}{=} s_{\text{eff}} \, L_r \, c \, C_t \overset{(7)}{=} l \, L \, c \, C_t \, / m \, , \tag{9}$$

and the number $N_{\text{E,r}}$ of entities in the system is

$$N_{\text{E,r}} = N_E \overset{(4)}{\approx} \frac{U}{\overline{e_s}} \overset{(9)}{\approx} \frac{U_r}{\overline{e_s}} \overset{(9)}{\approx} \frac{l \, L \, c \, C_t}{m \, \overline{e_s}} \, . \tag{10}$$

The system (referred to as *r-system* due to the redundancy introduced) comprises g_r *arrays* of m tapes, with each codeword stored across the tapes of an array. Thus,

$$g_r = \frac{L_r \, c}{m} \overset{(8)}{=} \frac{L \, c}{l} \, . \tag{11}$$

Each of the m tapes of an array is stored in a different library, which implies that $L_r \geq m$.

$$m \leq L_r \overset{(8)}{\iff} l \leq L \, . \tag{12}$$

Within an array, user data is stored in a *cluster* of l tapes as follows. The contents of each entity, such as Entity-1 and Entity-2, are divided into l *shards* that are stored in the l tapes of a cluster, as shown in Figure 1. In particular, the i-th ($i = 1, \ldots, l$) shard is stored in K symbols $S_{1,i}, S_{2,i}, \cdots, S_{K,i}$ with the fixed symbol size denoted by s and the data-set (symbol) boundaries indicated by the horizontal black lines in Figure 1(b). Then, the number C of data sets (symbols) in a tape is

$$C = \frac{C_t}{s} \, . \tag{13}$$

Subsequently, the l symbols $S_{j,1}, S_{j,2}, \cdots, S_{j,l}$ that correspond to the j-th ($1 \leq j \leq C$) symbol of each of the tapes in a cluster are complemented with $m - l$ parity symbols $S_{j,1+1}, \cdots, S_{j,m}$ to form codewords. To minimize the risk of permanent data loss, the $m-l$ parity symbols are stored in the remaining $m - l$ tapes of the array. This way, the system can tolerate any $\tilde{r} - 1$ tape failures, but $\tilde{r}$ tape failures may lead to data loss, with

$$\tilde{r} = m - l + 1 \, , \quad 1 \leq l < m \quad \text{and} \quad 2 \leq \tilde{r} \leq m \, . \tag{14}$$

The K codewords corresponding to Entity-1 are indicated by C-1, $\cdots$, C-K, as shown in Figure 1(b). For the convenience of illustration we depict codewords as aligned. From the above, it follows that the system contains g_r clusters and each cluster is associated with C codewords. The clusters and arrays are distributed across libraries such that within each library a fraction l/m of its tapes contain user data and the remaining tapes contain parity data. A relevant placement scheme is presented in detail in Appendix. The entire storage system is modeled as consisting of g_r independent arrays and clusters.

Note that one or more symbols are allocated to a shard with the first and last symbol potentially partially filled. The remaining space of a partially-filled symbol can be used to store the contents of another entity. User data is written sequentially such that a symbol may contain data of multiple entities. Therefore, shards and entities are stored in a way that is agnostic to symbol boundaries and therefore may not be aligned with symbols and codewords, respectively, as shown in Figure 1(b).

To access an entity, the l tapes of the corresponding cluster are mounted to l drives. Consequently, at any given time, there can be at most k_r entities processed, where

$$k_r = \frac{L_r \, d}{l} \overset{(8)}{=} \frac{m \, L \, d}{l^2} \, , \tag{15}$$

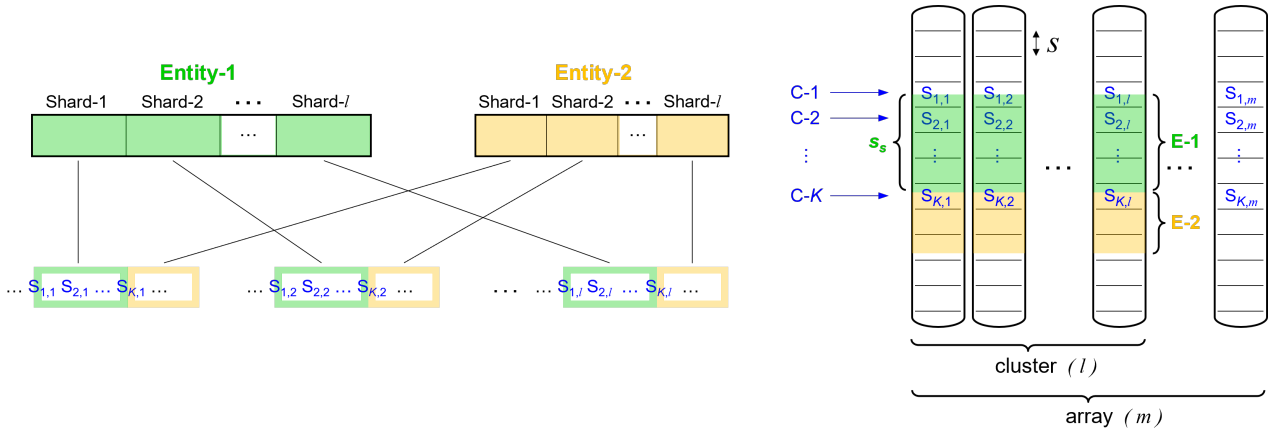
with each cluster mounted to one of the k_r tape drive groups. A tape-drive-group formation scheme that ensures the largest maximum entity throughput of the system is presented in Appendix. Also, the number $N_{\text{E,rc}}$ of entities in a cluster is [9, Eq.(14)]

$$N_{\text{E,rc}} \approx \frac{l \, C_t}{\overline{e_s}} \, . \tag{16}$$

III. Tape Library Operations

To perform read/write operations in a tape library system, tape cartridges are mounted to and unmounted from tape drives via an automated robotic mechanism. When a cartridge is mounted to a free drive, it is then loaded and, after a seek time, it is positioned to read/write the requested data. We consider an exhaustive service such that, upon completion of the read/write operations, the cartridge is rewound, unmounted and removed from the drive. Tapes are mounted according to a cyclic (round-robin) policy. To assess the effect of the various parameters, a performance model was developed and presented in [17][18]. As we are interested in evaluating the maximum throughput of the tape library system, we subsequently consider read operations only.

To serve requests for a tape, adopting the notation used in [17][18], it may take a waiting time W_{MT} for a robot arm to fetch the tape due to potential contention, a time R to mount it to the drive, a time t_L to load it, and a seek time s_I to start serving the initial request. Subsequent non-initial requests incur seek times s_N. As in [17][18], we consider a symmetric uniform random workload with entity requests arriving to the system at a rate of λ_e requests per unit of time. System operation is stable when $\lambda_e < \lambda_{e,\max}$, where $\lambda_{e,\max}$ is the maximum entity throughput of the system. When requests are served according to the First-Come-First-Served (FCFS) policy, the seek times $s_{I,\text{FCFS}}$ of initial requests are uniformly distributed in the interval [0,$s_{\max}$], where $s_{\max}$ is the maximum seek time corresponding to a request for the data located at the



(a) Creation of shards

(b) Generation and placement of codewords

Figure 1. Data placement of entities within a cluster, generation and placement of codewords within an array.

end of a tape. It therefore holds that the corresponding mean seek times $\overline{s_{I,\text{FCFS}}}$ and $\overline{s_{N,\text{FCFS}}}$ are given by [17, Eq.(2)], [18, Eq.(4)]

$$\overline{s_{I,\text{FCFS}}} = \frac{s_{\max}}{2} \quad \text{and} \quad \overline{s_{N,\text{FCFS}}} = \frac{s_{\max}}{3} . \tag{17}$$

The Recommended Access Order (RAO) policy [14] schedules the requests to be served in an order such that the seek times are reduced. In this case the seek times $s_{I,\text{RAO}}$ and $s_{N,\text{RAO}}$ depend on the system load, which in turn depends on the arrival rate of entity request λ_e. Therefore, it holds that

$$0 \le \overline{s_{I,\text{RAO}}}(\lambda_e) \le \overline{s_{I,\text{FCFS}}} , \quad 0 \le \overline{s_{N,\text{RAO}}}(\lambda_e) \le \overline{s_{N,\text{FCFS}}} . \tag{18}$$

In particular, for very low loads ($\lambda_e \approx 0$), scheduling a small number of requests does not in general significantly reduce the seek times, which implies that

$$\overline{s_{I,\text{RAO}}}(0^+) = \lim_{\lambda_e \to 0} \overline{s_{I,\text{RAO}}}(\lambda_e) = \overline{s_{I,\text{FCFS}}} , \tag{19}$$

and

$$\overline{s_{N,\text{RAO}}}(0^+) = \lim_{\lambda_e \to 0} \overline{s_{N,\text{RAO}}}(\lambda_e) = \overline{s_{N,\text{FCFS}}} . \tag{20}$$

Also, at very high loads ($\lambda_e \approx \lambda_{e,\max}$), the large number of requests can be optimally scheduled such that $s_I \approx 0$ and $s_N \approx 0$, which implies that

$$\overline{s_{I,\text{RAO}}}(\lambda^-_{e,\max}) = \lim_{\lambda_e \to \lambda_{e,\max}} \overline{s_{I,\text{RAO}}}(\lambda_e) = 0 , \tag{21}$$

and

$$\overline{s_{N,\text{RAO}}}(\lambda^-_{e,\max}) = \lim_{\lambda_e \to \lambda_{e,\max}} \overline{s_{N,\text{RAO}}}(\lambda_e) = 0 , \tag{22}$$

that is, under very high loads, the entire contents of the cartridge are read.

Note that various tape scheduling policies may be employed to accommodate efficiently data set profiles. For these policies it holds that

$$0 \le \overline{s_I}(\lambda_e) \le \overline{s_{I,\text{FCFS}}} \quad \text{and} \quad 0 \le \overline{s_N}(\lambda_e) \le \overline{s_{N,\text{FCFS}}} . \tag{23}$$

Depending on whether the system employs a reliability scheme to protect against data loss, two cases are considered (Sections III-A and III-B, respectively).

A. No Data Recovery and Protection

An entity of size e_s spans a number of K_n symbols with its expected value $E(K_n|e_s)$ obtained from (13) of [9] by setting $l = 1$ as follows:

$$E(K_n|e_s) = e_s/s + 1 . \tag{24}$$

Unconditioning on e_s yields

$$E(K_n) = \overline{e_s}/s + 1 . \tag{25}$$

Therefore, serving this arbitrary entity incurs a transfer time $t_T(e_s)$ determined by

$$t_T(e_s) = (K_n\, s)/b_w , \tag{26}$$

where b_w is the transfer bandwidth. Unconditioning on e_s, the mean value $\overline{t_T}$ of the transfer time t_T of an arbitrary entity is

$$\overline{t_T} = E(t_T(e_s)) = \frac{E(K_n)\, s}{b_w} \overset{(25)}{=} \frac{\overline{e_s} + s}{b_w} . \tag{27}$$

The total time to serve a request is the sum of the seek and transfer times. Therefore, the respective mean service times $\overline{B_I}$ and $\overline{B_N}$ of initial and non-initial requests are

$$\overline{B_I} = \overline{s_I} + \overline{t_T} \quad \text{and} \quad \overline{B_N} = \overline{s_N} + \overline{t_T} . \tag{28}$$

B. Data Recovery and Protection

A request for an entity of size e_s triggers l shard requests of size $s_s = e_s/l$ to each of the l tapes of the corresponding cluster. When the requests of this cluster are scheduled to be served, the l tapes of the cluster are mounted. Note that, as each of the l tapes resides in a different library, each of these tapes is mounted and, subsequently, unmounted by a different robot arm. The number of codewords, K, that this entity spans or, equivalently, the number of symbols that a corresponding shard spans, is obtained from (13) of [9] as follows:

$$E(K|e_s) = e_s/(l\, s) + 1 . \tag{29}$$

Unconditioning on e_s yields

$$E(K) = \overline{e_s}/(l\, s) + 1 . \tag{30}$$

Therefore, serving this entity incurs a transfer time $t_{\text{T,r}}(e_s)$ determined by

$$t_{\text{T,r}}(e_s) = (K\, s)/b_w . \tag{31}$$

Unconditioning on e_s, the mean value $\overline{t_{\text{T,r}}}$ of the transfer time $t_{\text{T,r}}$ of an arbitrary entity is

$$\overline{t_{\text{T,r}}} \;=\; E(t_{\text{T,r}}(e_s)) \;=\; \frac{E(K)\,s}{b_w} \;\overset{(30)}{=}\; \frac{\overline{e_s} + l\,s}{l\,b_w}\;. \tag{32}$$

The total time to serve a request is the sum of the seek and transfer times. Therefore, the respective mean service times $\overline{B_{\text{I,r}}}$ and $\overline{B_{\text{N,r}}}$ of initial and non-initial requests are

$$\overline{B_{\text{I,r}}} = \overline{s_I} + \overline{t_{\text{T,r}}} \qquad \text{and} \qquad \overline{B_{\text{N,r}}} = \overline{s_N} + \overline{t_{\text{T,r}}}\;. \tag{33}$$

IV. Maximum Throughput

The maximum throughput is achieved when the system is at saturation, that is, when $\lambda_e = \lambda_{e,\max}$. First, we consider a sequential workload and then a symmetric uniform random workload (Sections IV-A and IV-B, respectively).

A. Sequential Workload

We assume that mounted tapes are read in their entirety.

1) No Data Recovery and Protection: The time T_c required to read the $N_{\text{E,c}}$ entities stored in a tape is

$$T_c \;=\; C_t/b_w\;. \tag{34}$$

Considering the time to mount and unmount a tape $M+U$, the maximum entity throughput of a tape drive is $N_{\text{E,c}}/(T_c + \overline{M} + \overline{U})$. As there are k drives in the system, the maximum entity throughput $\lambda_{e,\max}$ of the system is then determined by

$$\lambda_{e,\max} \;=\; k\,\frac{N_{\text{E,c}}}{T_c + \overline{M} + \overline{U}}\;. \tag{35}$$

Today, in practice, the mean time to mount and unmount a tape $\overline{M} + \overline{U}$ (which is typically in the order of seconds) is negligible compared with T_c (which is typically in the order of hours). Consequently, from (35), it follows that

$$\lambda_{e,\max} \;\approx\; k\,\frac{N_{\text{E,c}}}{T_c} \;\overset{(5)(6)(34)}{\approx}\; \frac{L\,d\,b_w}{\overline{e_s}}\;. \tag{36}$$

According to (22), for the RAO policy, it holds that

$$\lambda_{e,\max} \;\approx\; \frac{L\,d\,b_w}{\overline{e_s}}\;, \qquad \text{for RAO.} \tag{37}$$

2) Data Recovery and Protection: In this case, the time T_c is the time required to read the $N_{\text{E,rc}}$ entities stored in a cluster. Therefore, the maximum entity throughput of a tape drive group is roughly $N_{\text{E,rc}}/T_c$. As there are k_r tape drive groups in the system, the maximum entity throughput $\lambda_{e,\max}$ of the system is then determined by

$$\lambda_{e,\max} \;=\; k_r\,\frac{N_{\text{E,rc}}}{T_c + \overline{M} + \overline{U}}\;. \tag{38}$$

Considering the mean time to mount and unmount a tape $\overline{M} + \overline{U}$ to be negligible compared with T_c, from (38), it follows that

$$\lambda_{e,\max} \;\approx\; k_r\,\frac{N_{\text{E,rc}}}{T_c} \;\overset{(15)(16)(34)}{\approx}\; \frac{m\,L\,d\,b_w}{l\,\overline{e_s}}\;. \tag{39}$$

According to (22), for the RAO policy, it holds that

$$\lambda_{e,\max} \;\approx\; \frac{m\,L\,d\,b_w}{l\,\overline{e_s}}\;, \qquad \text{for RAO.} \tag{40}$$

Clearly, in both cases, the seek times $s_{s,N,\text{RAO}}$ for the RAO policy at saturation are 0, that is,

$$s_{s,N,\text{RAO}} = \overline{s_{s,N,\text{RAO}}} \;=\; \overline{s_{N,\text{RAO}}}\,(\lambda_{e,\max}^{-}) \;\overset{(22)}{=}\; 0\;. \tag{41}$$

B. Symmetric Uniform Random Workload

Let us denote by $s_{\text{s,N}}$ and $B_{\text{s,N}}$ the seek and service times of non-initial requests at saturation, respectively, that is

$$s_{\text{s,N}} \triangleq s_N(\lambda_{e,\max}) \quad \text{and} \quad B_{\text{s,N}} \triangleq B_N(\lambda_{e,\max})\;. \tag{42}$$

Subsequently, the maximum throughput of entities served by a tape drive is $1/\,\overline{B_{\text{s,N}}}$. As there are k tape drives in the system, the maximum entity throughput $\lambda_{e,\max}$ of the system is then determined by

$$\lambda_{e,\max} = \frac{k}{\overline{B_{\text{s,N}}}}\;, \tag{43}$$

where $\overline{B_{\text{s,N}}}$ is obtained from (28) as follows:

$$\overline{B_{\text{s,N}}} = \overline{s_{\text{s,N}}} + \overline{t_T} \tag{44}$$

where $\overline{s_{\text{s,N}}}$ is the mean seek time of non-initial requests at saturation.

Also, let $s_{s,N,\text{FCFS}}$ denote the seek times of non-initial requests at saturation for the FCFS policy. Then, for a symmetric uniform random workload we have

$$\overline{s_{s,N,\text{FCFS}}} \;=\; \overline{s_{N,\text{FCFS}}} \;\overset{(17)}{=}\; \frac{s_{\max}}{3}\;. \tag{45}$$

Note that the seek times $s_{\text{s,N}}$ may in general correspond to a scheduling policy that, at saturation, schedules the requests to be served in a non-sequential order and with the seek times being reduced compared to the FCFS policy. It may also represent reduced seek times due to technological advancements of next generation cartridges. We subsequently consider $\overline{s_{\text{s,N}}}$ to take values in the range $[0, \overline{s_{s,N,\text{FCFS}}}]$, that is,

$$0 \le \overline{s_{\text{s,N}}} \le \overline{s_{s,N,\text{FCFS}}}\;. \tag{46}$$

Remark 1: When $s_{\text{s,N}} = \overline{s_{\text{s,N}}} = 0$, the maximum throughput is smaller than that achieved by RAO, for which, according to (41), it also holds that $\overline{s_{s,N,\text{RAO}}} = 0$. This is due to the fact that when entity requests are not served sequentially, data sets on the entity boundaries are read twice. Consequently, the smaller the entity size, the more pronounced the difference between these two maximum throughputs.

Depending on whether redundancy is introduced to protect the system against data loss, two cases are considered.

1) No Data Recovery and Protection: Substituting (6) and (44) into (43), and using (27), yields the maximum entity throughput $\lambda_{e,\max}$ as follows:

$$\lambda_{e,\max} = \frac{L\,d\,b_w}{\overline{e_s} + s + b_w\;\overline{s_{\text{s,N}}}}\;. \tag{47}$$

In particular, for the FCFS policy, using (45) we get

$$\lambda_{e,\max} = \frac{L\,d\,b_w}{\overline{e_s} + s + b_w\;s_{\max}/3}\;, \qquad \text{for FCFS.} \tag{48}$$

Remark 2: When $\overline{e_s} \gg s$, from (47) and (48), it follows that

$$\lambda_{e,\max} \;\approx\; \frac{L\,d\,b_w}{\overline{e_s} + b_w\;\overline{s_{\text{s,N}}}}\;, \tag{49}$$

and

$$\lambda_{e,\max} \;\approx\; \frac{L\,d\,b_w}{\overline{e_s} + b_w\;\dfrac{s_{\max}}{3}}\;, \qquad \text{for FCFS.} \tag{50}$$

Remark 3: When $\overline{e_s} \gg s$, from (37) and (50), it follows that employing the RAO policy results in an increase of the FCFS maximum throughput by a factor of f_{RAO} determined by

$$f_{\text{RAO}} \triangleq \frac{\lambda_{e,\max}(\text{RAO})}{\lambda_{e,\max}(\text{FCFS})} \approx 1 + \frac{b_w}{\overline{e_s}} \cdot \frac{s_{\max}}{3} . \quad (51)$$

The above implies that the smaller the average entity size $\overline{e_s}$, the larger the factor f_{RAO}.

2) Data Recovery and Protection: As there are k_r tape drive groups in the system and the maximum throughput of entities served by a tape drive group is $1/\overline{B_{\text{s,N,r}}}$, it follows that the maximum entity throughput $\lambda_{e,\max}$ of the system is determined by

$$\lambda_{e,\max} = \frac{k_r}{\overline{B_{\text{s,N,r}}}} , \quad (52)$$

where, by virtue of (33) and (44), $\overline{B_{\text{s,N,r}}}$ is obtained by

$$\overline{B_{\text{s,N,r}}} = \overline{s_{\text{s,N}}} + \overline{t_{\text{T,r}}} . \quad (53)$$

Substituting (15) and (53) into (52), and using (32), yields the maximum entity throughput $\lambda_{e,\max}$ as follows:

$$\lambda_{e,\max} = \frac{L_r\, d\, b_w}{\overline{e_s} + l\,(s + b_w\ \overline{s_{\text{s,N}}})} = \frac{m\, L\, d\, b_w}{l\,[\overline{e_s} + l\,(s + b_w\ \overline{s_{\text{s,N}}})]} . \quad (54)$$

In particular, for the FCFS policy, using (45) we get

$$\lambda_{e,\max} = \frac{m\, L\, d\, b_w}{l\,[\overline{e_s} + l\,(s + b_w\ s_{\max}/3)]} , \quad \text{for FCFS.} \quad (55)$$

Remark 4: When $\overline{e_s} \gg l\, s$, from (54) and (55), it follows that

$$\lambda_{e,\max} \approx \frac{L_r\, d\, b_w}{\overline{e_s} + l\, b_w\ \overline{s_{\text{s,N}}}} \approx \frac{m\, L\, d\, b_w}{l\,[\overline{e_s} + l\, b_w\ \overline{s_{\text{s,N}}}]} , \quad (56)$$

and

$$\lambda_{e,\max} \approx \frac{m\, L\, d\, b_w}{l\,(\overline{e_s} + l\, b_w\ s_{\max}/3)} , \quad \text{for FCFS.} \quad (57)$$

Remark 5: For r-systems of fixed storage efficiency s_{eff}, from (54), it follows that the maximum entity throughput corresponding to the mean seek time $\overline{s_{\text{s,N}}}$ of non-initial requests at saturation is decreasing in l. Also, for the RAO policy and according to (40), we deduce that the corresponding maximum throughputs are roughly the same.

Remark 6: From (56) and (57), it follows that the maximum throughput for the policy corresponding to the mean seek time $\overline{s_{\text{s,N}}}$ of non-initial requests at saturation is larger that of the FCFS policy by a factor $f_{\overline{s_{\text{s,N}}}}$ determined by

$$f_{\overline{s_{\text{s,N}}}} \triangleq \frac{\lambda_{e,\max}(\overline{s_{\text{s,N}}})}{\lambda_{e,\max}(\overline{s_{N,\text{FCFS}}})} \approx \frac{\overline{e_s} + l\,(s + b_w\ s_{\max}/3)}{\overline{e_s} + l\,(s + b_w\ \overline{s_{\text{s,N}}})} . \quad (58)$$

The above implies that the factor $f_{\overline{s_{\text{s,N}}}}$ depends on l, but not on the codeword length m.

Remark 7: When $\overline{e_s} \gg l\, s$, from (40) and (57), it follows that employing the RAO policy results in an increase of the FCFS maximum throughput by a factor of f_{RAO} determined by

$$f_{\text{RAO}} \triangleq \frac{\lambda_{e,\max}(\text{RAO})}{\lambda_{e,\max}(\text{FCFS})} \approx 1 + l \cdot \frac{b_w}{\overline{e_s}} \cdot \frac{s_{\max}}{3} . \quad (59)$$

TABLE II. PARAMETER VALUES

Parameter	Definition	Values
c	number of tape cartridges	3200
d	number of tape drives	32
a	number of robot arms	1, 2
R	robot transfer time	5 s (fixed)
t_L	load ready time	15 s (fixed)
t_U	unload ready time	24 s (fixed)
$s_{\max}$	maximum seek time	118 s
$\overline{e_s}$	mean request size	843 MB
b_w	bandwidth	360 MB/s
$\overline{t_R}$	mean rewind time for random workload	59 s
M	mount time ($M = R + t_L$)	20 s (fixed)
$\overline{U}$	mean unmount time ($\overline{U} = \overline{t_R} + t_U + R$)	88 s
t_T	mean transfer time ($t_T = \overline{e_s}/b_w$)	2.34 s

The above implies that the larger the number of user-data symbols per codeword l and the smaller the average entity size $\overline{e_s}$, the larger the factor f_{RAO}.

Remark 8: From (47), it follows that the maximum entity throughput $\lambda_{e,\max}$ of the original unprotected system can be obtained from (54) by setting $m = l = 1$.

Remark 9: For the FCFS policy, from (48) and (55), it follows that the maximum entity throughput of an r-system is greater than or equal to that of the original system when

$$\overline{e_s} \geq \frac{l^2 - m}{m - l} \left(s + b_w \frac{s_{\max}}{3}\right) . \quad (60)$$

Remark 10: For the RAO policy, from (37) and (40), and given that $m > l$, it follows that the maximum entity throughput of an r-system is greater than that of the original one.

Remark 11: When $m < l^2$, from (47) and (54), it follows that the maximum throughput of the r-system is greater than or equal to that of the original system when $\overline{s_{\text{s,N}}}$ is less than or equal to $s^*_{\text{s,N}}$ determined by

$$s^*_{\text{s,N}} = \frac{(m - l)\ \overline{e_s} - (l^2 - m)\, s}{(l^2 - m)\, b_w} , \quad (61)$$

which, when $\overline{e_s} \gg s$, reduces to

$$s^*_{\text{s,N}} \approx \frac{m - l}{l^2 - m} \cdot \frac{\overline{e_s}}{b_w} . \quad (62)$$

Consequently, the maximum throughputs of J r-systems, with each one employing a different MDS(m_j,l_j) erasure code ($1 \leq j \leq J$), are equal to that of the original system at the same $\overline{s_{\text{s,N}}}$ value, when the following condition holds:

$$\frac{m_1 - l_1}{l_1^2 - m_1} = \cdots = \frac{m_j - l_j}{l_j^2 - m_j} = \cdots = \frac{m_J - l_J}{l_J^2 - m_J} . \quad (63)$$

Remark 12: From (37), (40), (47), and (54), it follows that the maximum entity throughput $\lambda_{e,\max}$ is insensitive to the entity size distribution, that is, it depends only on its average $\overline{e_s}$ determined by (2), but not on its density $\{v_j\}$ given by (1).

Remark 13: According to the discussion in [17][18], the maximum entity throughput $\lambda_{e,\max}$ is the same for both the *Always-Unmount* (AU) and *Not-Unmount* (NU) policies given that at high loads, when all requests for a given tape are served, there are pending requests and therefore the tape is unmounted.

TABLE III. MAXIMUM THROUGHPUT (entities/second).

MDS Protection Scheme	L	L_r	CERN File Size Distribution		
			FCFS	RAO	f_{RAO}
none	6		4.605	81.509	17.803
MDS(2,1)	6	12	9.211	163.019	17.803
MDS(3,2)	6	9	3.553	121.547	34.606
MDS(4,2)	6	12	4.738	162.063	34.606
MDS(4,3)	6	8	2.118	106.993	51.409
MDS(6,3)	6	12	3.189	161.119	51.409
MDS(6,4)	6	9	1.803	120.139	68.212

V. NUMERICAL RESULTS

Here, we consider a system comprised of $L = 6$ IBM TS4500 tape libraries [19]. In particular, we consider a 4-frame library configuration for the parameter values listed in Table 2 of [18], which is reproduced here in Table II for completeness. Cartridges and drives correspond to the LTO-8 tape technology with $C_t = 12$ TB [16]. Each library comprises $d = 32$ tape drives and $c = 3200$ cartridges with a maximum seek time $s_{\max}$ of 118 s (for LTO-8 full high drive [2]). From (45), it follows that the mean seek time of non-initial requests at saturation for the FCFS policy is $\overline{s_{s,N,\text{FCFS}}} = 118/3 = 39.33$ s. The robot access times are fixed equal to $R = 5$ s, the mount times are fixed equal to $M = 20$ s, and the mean unmount time is $U = 88$ s.

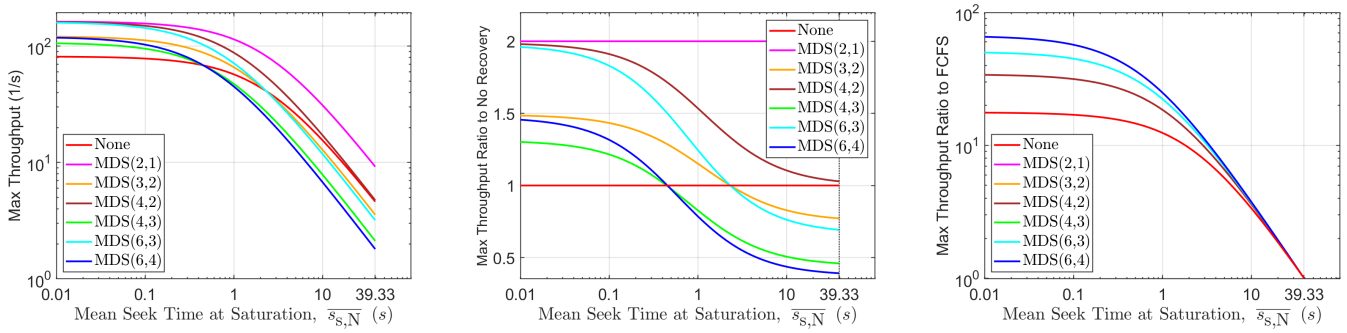
We consider a system storing files with a size distribution as described by CERN and given in [7][8], such that their average size $\overline{e_s}$ is 843 MB. Subsequently, we assess the maximum system throughput assuming a symmetric uniform random workload and considering the entire range of the mean seek time $\overline{s_{\text{s,N}}}$ of non-initial requests at saturation between the two extremes corresponding to the RAO and FCFS policies. The maximum throughput as a function of the mean seek time $\overline{s_{\text{s,N}}}$ is obtained from (47) and shown by the red line in Figure 2(a). The maximum throughputs corresponding to the FCFS and RAO policies are 4.61 and 81.99 (entity requests per second), respectively, as summarized in Table III. Although we consider $\overline{s_{\text{s,N}}}$ to take values in the range $[0, \overline{s_{s,N,\text{FCFS}}}]$, that is, $[0, 39.33]$, owing to the logarithmic x-axis, we show results starting at $\overline{s_{\text{s,N}}} = 0.01$ s and not at $\overline{s_{\text{s,N}}} = 0$ s. We observe that employing the RAO policy results in an increase of the FCFS maximum throughput by a factor of $f_{\text{RAO}} = 81.509/4.605 = 17.803$, as summarized in Table III, which is close to the value of 17.623 indicated by the red line in Figure 2(c) at $\overline{s_{\text{s,N}}} = 0.01$ and the value of 17.698 at $\overline{s_{\text{s,N}}} = 0$ (not shown).

We subsequently consider r-systems employing the following erasure coding schemes: (2,1), (3,2), (4,2), (4,3), (6,3), and (6,4) MDS erasure codes. The (2,1) MDS erasure coding scheme corresponds to a system with replication factor of two. The (2,1), (4,2) and (6,3) MDS erasure codes have a storage efficiency of 50%, which requires $L_r = 12$ libraries; the (3,2) and (6,4) MDS erasure codes have a storage efficiency of 66%, which requires $L_r = 9$ libraries; and the (4,3) MDS erasure code has a storage efficiency of 75%, which requires $L_r = 8$ libraries, as summarized in Table III.

The maximum entity throughput $\lambda_{e,\max}$ as a function of the mean seek time $\overline{s_{\text{s,N}}}$ for the various MDS codes is shown in Figure 2(a). From (47) and (56), it follows that the maximum throughputs are strictly decreasing in $\overline{s_{\text{s,N}}}$. According to Remark 5, the maximum throughput of the MDS(3,2) r-system is larger than that of the MDS(6,4) one. In particular, for the RAO policy, the corresponding maximum throughputs are roughly the same equal to 121.547 and 120.139, respectively, as listed in Table III. For the FCFS policy, at $\overline{s_{\text{s,N}}} = 39.33$, the maximum throughputs are 3.553 and 1.803 for the MDS(3,2) and MDS(6,4) r-systems, respectively. Similarly, the maximum throughput of the MDS(2,1) r-system is larger than that of the MDS(4,2) r-system, which in turn is larger than that of the MDS(6,3) one. For the RAO policy, the corresponding maximum throughputs are roughly the same equal to 163.019, 162.063, and 161.119, respectively. For the FCFS policy, at $\overline{s_{\text{s,N}}} = 39.33$, the maximum throughputs are 9.211, 4.738, and 3.189 for the MDS(2,1), MDS(4,2), and MDS(6,3) r-systems, respectively.

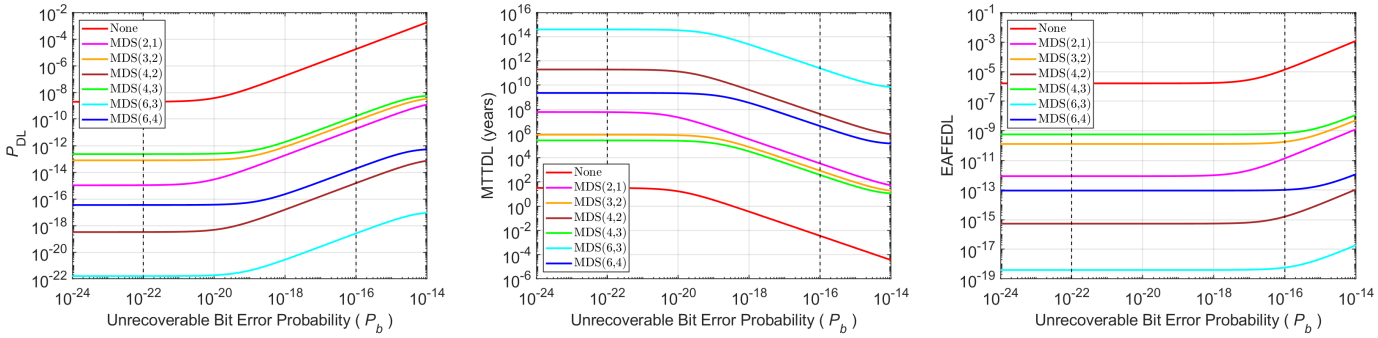
The corresponding ratios of the maximum throughputs of the r-systems to the maximum throughput of the original system are plotted in Figure 2(b). According to Remark 11, the maximum throughput of an r-system is roughly equal to that of the original system when $\overline{s_{\text{s,N}}} = s^*_{\text{s,N}}$ as determined by (62). Moreover, given that the MDS(4,3) and MDS(6,4) erasure codes satisfy condition (63), the maximum throughputs of these r-systems are equal to that of the original system at the same $\overline{s_{\text{s,N}}}$ value, as shown in Figures 2(a) and 2(b). The same holds for the the MDS(3,2) and MDS(6,3) r-systems. As a consequence, the FCFS maximum throughputs of these MDS r-systems are smaller than that of the original system. This, however, does not hold for the MDS(2,1) and MDS(4,2) r-systems, because, according to Remark 9, and given that (60) holds, the resulting FCFS maximum throughputs are larger than that of the original system. In fact, we observe that the maximum throughputs of the MDS(2,1) and MDS(4,2) r-systems are larger than that of the original system in the entire range of $\overline{s_{\text{s,N}}}$. In particular, the maximum throughput of the MDS(2,1) r-system is twice as large as that of the original system. Furthermore, according to Remark 6, the improvement factors $f_{\overline{s_{\text{s,N}}}}$ of the maximum throughputs for the MDS erasure codes considered over the FCFS one depend on l, but not on the codeword length m. Consequently, the $f_{\overline{s_{\text{s,N}}}}$ factors for the MDS(3,2) and MDS(4,2) r-systems are the same and therefore, in Figure 2(c), the orange line for the MDS(3,2) r-system is not visible because it lies just below the brown one for the MDS(4,2) r-system. The same holds for the MDS(4,3) and MDS(6,3) r-systems and therefore the green line lies below the cyan one. Also, the magenta line corresponding to the MDS(2,1) r-system lies below the red one corresponding to the original system. In particular, the f_{RAO} factors for the MDS(3,2) and MDS(4,2) r-systems are the same and equal to 34.606, as listed in Table III. The same holds for the MDS(4,3) and MDS(6,3) r-systems where $f_{\text{RAO}} = 51.409$. Also, for the MDS(2,1) r-system and the original system, we have $f_{\text{RAO}} = 17.803$.

Next, we assess the reliability of the systems considered assuming a symmetric uniform random workload. Entity requests arrive according to a Poisson process at a rate of $\lambda_e = 0.5$ entity requests per second and are considered to be served according to the FCFS policy. A permanent data loss may occur following the mounting of l tapes of a cluster to a drive group and during subsequent operations on that group. The probability P_{DL} of data loss as well as the MTTDL and EAFEDL reliability metrics are evaluated using the results



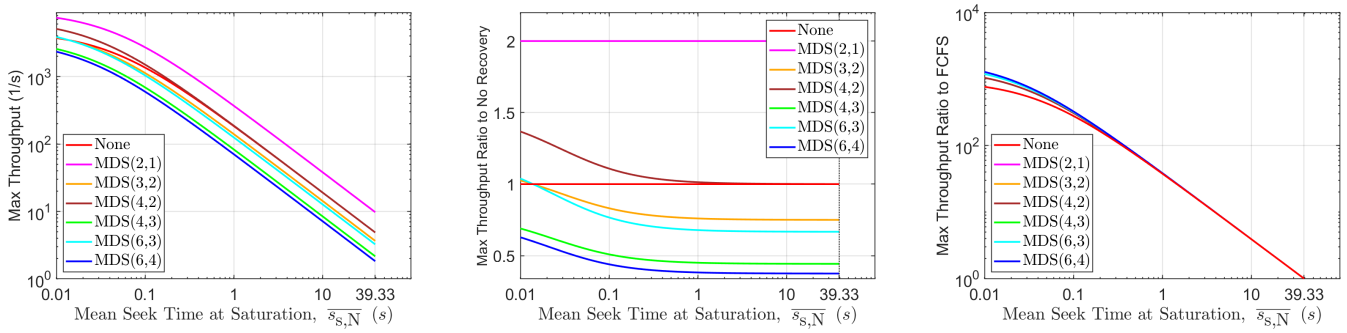
(a) Maximum throughput, $\lambda_{e,\max}$ (b) MDS to no recovery maximum throughput ratio (c) RAO to FCFS maximum throughput ratio, f_{RAO}

Figure 2. Maximum throughput vs. mean seek time for systems that employ various MDS erasure codes; $L = 6$, $d = 32$, CERN file size distribution.



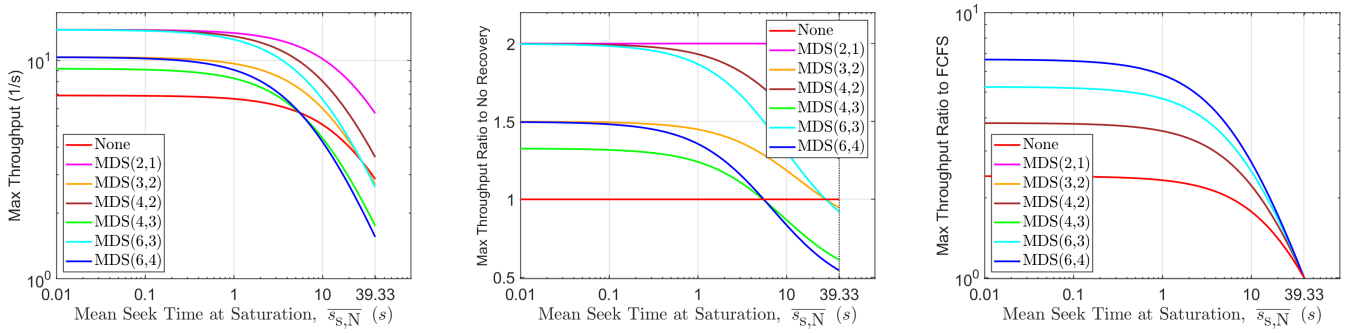
(a) Probability of data loss, P_{DL} (b) MTTDL (c) EAFEDL

Figure 3. Reliability measures vs. bit error rate for various MDS erasure codes.



(a) Maximum throughput, $\lambda_{e,\max}$ (b) MDS to no recovery maximum throughput ratio (c) RAO to FCFS maximum throughput ratio, f_{RAO}

Figure 4. Maximum throughput vs. mean seek time for systems that employ various MDS erasure codes; $L = 6$, $d = 32$, $\overline{e_s} = 10$ MB.



(a) Maximum throughput, $\lambda_{e,\max}$ (b) MDS to no recovery maximum throughput ratio (c) RAO to FCFS maximum throughput ratio, f_{RAO}

Figure 5. Maximum throughput vs. mean seek time for systems that employ various MDS erasure codes; $L = 6$, $d = 32$, $\overline{e_s} = 10$ GB.

TABLE IV. MAXIMUM THROUGHPUT (entities/second).

MDS Protection Scheme	L	L_r	Mean File Size = 10 MB		
			FCFS	RAO	f_{RAO}
none	6		4.876	6,912	1,417.50
MDS(2,1)	6	12	9.752	13,824	1,417.50
MDS(3,2)	6	9	3.658	10,368	2,834.00
MDS(4,2)	6	12	4.877	13,824	2,834.00
MDS(4,3)	6	8	2.159	9,216	4,267.17
MDS(6,3)	6	12	3.252	13,824	4,250.50
MDS(6,4)	6	9	1.829	10,368	5,667.00

TABLE V. MAXIMUM THROUGHPUT (entities/second).

MDS Protection Scheme	L	L_r	Mean File Size = 10 GB		
			FCFS	RAO	f_{RAO}
none	6		2.860	6.912	2.416
MDS(2,1)	6	12	5.720	13.824	2.416
MDS(3,2)	6	9	2.704	10.368	3.833
MDS(4,2)	6	12	3.606	13.824	3.833
MDS(4,3)	6	8	1.748	9.216	5.270
MDS(6,3)	6	12	2.633	13.824	5.249
MDS(6,4)	6	9	1.555	10.368	6.666

presented in [9] and shown in Figure 3. We observe that the (4,3) MDS code, with a corresponding MDS-code distance $\tilde{r}$ of 2, yields the lowest reliability, which can be successively improved by employing the (3,2), (2,1), (6,4), (4,2), and (6,3) MDS codes with the corresponding MDS-code distances $\tilde{r}$ of 2, 2, 3, 3, and 4, respectively. Clearly, increasing the MDS-code distance improves reliability and for the codes that have the same distance, the ones with smaller codeword lengths or, equivalently, lower storage efficiencies achieve a higher reliability. Note that the r-system that yields the largest maximum throughput is the MDS(2,1) one, but the r-system that yields the highest reliability is the MDS(6,3) one. These results suggest that there is a tradeoff between reliability and performance, which will be the subject of future work.

The effect of the mean entity size on the maximum throughput is assessed by considering the cases of $\overline{e_s} = 10$ MB and $\overline{e_s} = 10$ GB shown in Figures 4 and 5. The corresponding maximum entity throughput values for the FCFS and RAO policies are listed in Tables IV and V, respectively. We observe that the order of the relative throughputs of the various r-systems remains the same. Also, when the mean entity size increases, the knees of the curves shift to the right. In the case of $\overline{e_s} = 10$ MB, the maximum throughputs for RAO are 1000 times larger than the corresponding ones when $\overline{e_s} = 10$ GB. This is due to the fact that, in the case of $\overline{e_s} = 10$ GB, a tape contains 1000 times more entities, which is the ratio of the mean entity sizes. Also, the improvement compared to the FCFS policy is more pronounced for smaller entity sizes.

In the case of $\overline{e_s} = 10$ MB, the maximum throughputs at $\overline{s_{\text{s,N}}} = 0$ are significantly lower than those of the RAO policy. As mentioned in Remark 1, this is due to the fact that data sets are read repeatedly. In the original system, the maximum throughput at $s_{\text{s,N}} = 0$ is 4608, whereas the maximum throughput for RAO is 6912, which is 1.5 times larger. Note that for a symbol (data set) size of 5 MB, a stored entity of size 10 MB spans 3 symbols, because it is not aligned with symbol boundaries. From (25), it follows that an arbitrary entity spans on the average $10/5 + 1 = 3$ symbols, too. Therefore, accessing an entity entails reading on the average $3 \cdot 5 = 15$ MB, resulting in an an overhead of $15/10 = 1.5$, which also reflects the difference in the maximum throughputs. Similarly, for the MDS(6,4) r-system, the maximum throughput at $s_{\text{s,N}} = 0$ is 3,456, whereas the maximum throughput for RAO is 10,368, which is 3 times larger. Note that a stored entity of size 10 MB corresponds to shards of size $10/4 = 2.5$ MB. As these shards are not aligned with symbols, they span either one symbol with probability 0.5 or two symbols with probability 0.5, for an average of 1.5 symbols. From (30), it follows that an arbitrary shard spans on the average $2.5/5+1 = 1.5$ symbols, too. Therefore, accessing a shard entails reading on the average $1.5 \cdot 5 = 7.5$ MB, resulting in an increased overhead of $7.5/2.5 = 3$, which also reflects the difference in the maximum throughputs.

VI. CONCLUSIONS

Tape library systems may experience cartridge damages and data losses due to power and operational failures. To cope with this issue, tape systems can be protected through the employment of erasure-coding redundancy schemes. The effectiveness of these schemes and the corresponding reliability of automated tape library systems has been evaluated based on the Mean Time to Data Loss (MTTDL) and the Expected Annual Fraction of Entity Loss (EAFEDL) metric, which assesses data losses at an entity, say file, object, or block level. The maximum throughput of the system is affected by the capability of the erasure coding scheme employed and it also depends on the scheduling policy implemented and the characteristics of the workload considered. The maximum throughput was obtained analytically in closed form for erasure-coding redundancy schemes and for a spectrum of scheduling policies including the First-Come-First-Served (FCFS) and the Recommended Access Order (RAO) policy. It was demonstrated that employing erasure coding may adversely affect the maximum throughput. We established that the maximum throughput improves when requests are served according to schemes, such as RAO, that effectively reduce seek times. The analytical results obtained enable the identification of erasure-coded redundancy schemes that ensure an acceptable system processing capability, which is reflected by the maximum throughput, and a desired level of reliability.

This work has the potential to be applied for further studies of tape storage performance and it is particularly relevant for assessing the effect of various system configurations and enabling a better understanding of the design tradeoffs between performance and reliability.

APPENDIX

Here, we present a scheme to store user and parity data in a way that ensures the largest maximum entity throughput of an r-system that employs an MDS(m,l) erasure code and comprises L_r tape libraries, with each one containing d tape drives and c tape cartridges.

First, we describe the formation of the k_r tape drive groups, as shown in Figure 6, for a system with $L_r = 9$, $d = 8$ tape drives (indicated by the circles), and $l = 4$, such that $k_r = L_r\, d/l = 9 \cdot 8/4 = 18$ drive groups. Groups of $l = 4$ drives are formed successively by starting in the first line, continuing in the second line, and so forth, until the last ($d = 8$-th) line. In

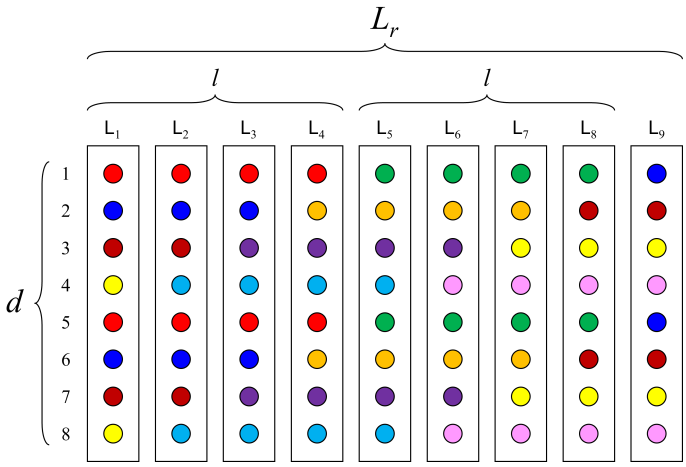


Figure 6. Formation of the $k_r = 18$ tape drive groups in an MDS(m, $l = 4$) r-system comprised of $L_r = 9$ libraries and $d = 8$ tape drives.

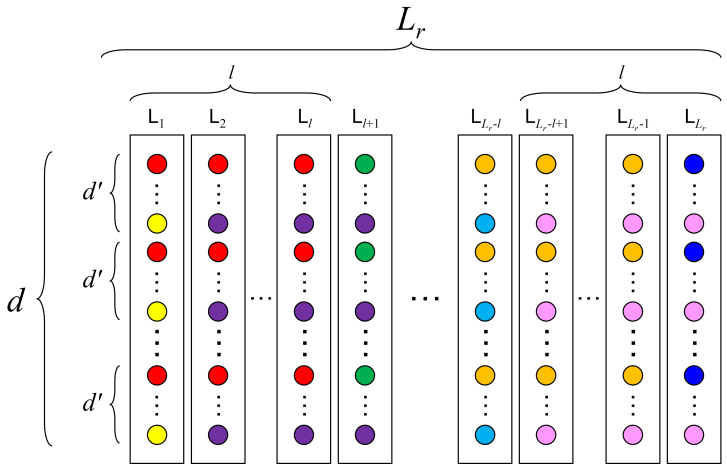


Figure 7. Formation of the k_r tape drive groups in an MDS(m,l) r-system comprised of L_r libraries and d tape drives.

Figure 6, the first line contains two drive groups, namely, the red and the green one. The red group comprises four drives in the $S_1 = \{\mathrm{L}_1, \mathrm{L}_2, \mathrm{L}_3, \mathrm{L}_4\}$ subset of libraries. Next, the green group comprises drives in the $S_2 = \{\mathrm{L}_5, \mathrm{L}_6, \mathrm{L}_7, \mathrm{L}_8\}$ subset of libraries. The third (blue) group is formed by continuing in the second line and comprises drives in the $S_3 = \{\mathrm{L}_9, \mathrm{L}_1, \mathrm{L}_2, \mathrm{L}_3\}$ subset of libraries. The first four lines contain nine drive groups with each one involving a different subset of four libraries. The remaining four lines contain the remaining nine groups. In general, let d' be the minimum number of lines required such that the last group in the d'-th line comprises drives in the subset $\{\mathrm{L}_{L_r-l+1}, \mathrm{L}_{L_r-l+2}, \cdots, \mathrm{L}_{L_r-1}, \mathrm{L}_{L_r}\}$ of libraries, as shown in Figure 7. Then, it holds that

$$d' = \frac{LCM(L_r, l)}{L_r} , \tag{64}$$

where $LCM(x, y)$ denotes the lowest common multiple of x and y. Note that each of the drive groups in the d' lines involves a different subset of libraries. The number k' of these drive groups is determined by

$$k' = \frac{LCM(L_r, l)}{l} . \tag{65}$$

In the example considered in Figure 6, it holds that $d' = LCM(9, 4)/9 = 36/9 = 4$ and $k' = LCM(9, 4)/4 = 9$.

The tape-drive-group formation scheme described may result in multiple groups with the corresponding tape drives located in the same subset of l libraries. In Figure 6, drive groups of the same color comprise drives in the same subset of libraries. Let $S = \{S_1, S_2, \cdots, S_i, \cdots, S_{k'}\}$ be the set of the k' subsets of libraries. In each of these subsets correspond k^* of the k_r drive groups, where k^* is determined by

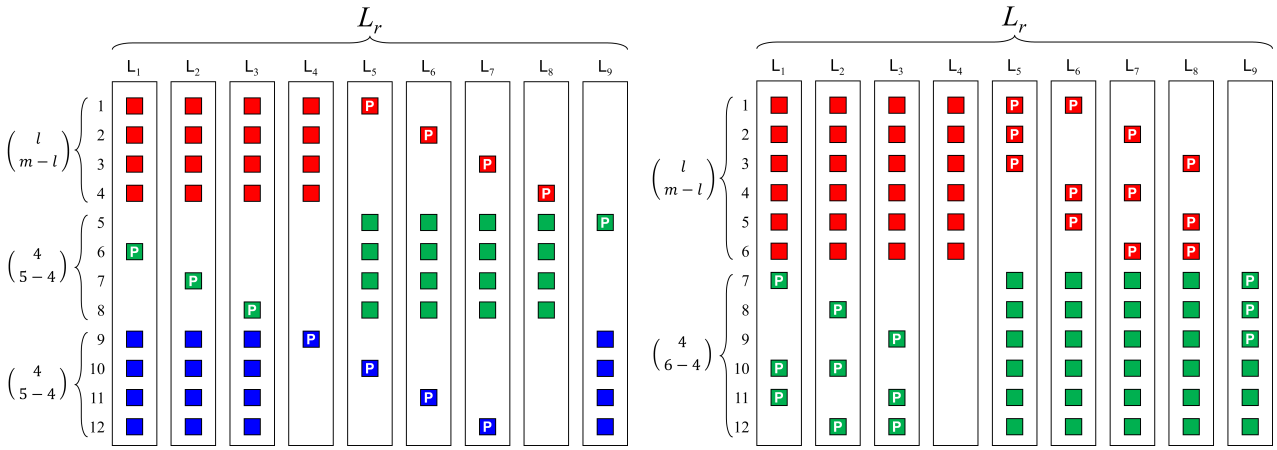
$$k^* = \frac{k_r}{k'} \overset{(15)(65)}{=} \frac{L_r\, d}{LCM(L_r, l)} \leq d . \tag{66}$$

In the example considered in Figure 6, it holds that $k^* = 9 \cdot 8/LCM(9, 4) = 72/36 = 2$, which implies that there are 2 drive groups of any given color. As there are g_r clusters in the system, the number g^* of clusters corresponding to any given of the k' subsets of libraries is

$$g^* = \frac{g_r}{k'} , \tag{67}$$

where g_r and k' are determined by (11) and (65), respectively.

Next, we describe the way that user data and parity data is stored in the system. To store an entity, its contents are divided



(a) MDS(5,4) (b) MDS(6,4)

Figure 8. Placement of the user and parity data in MDS(m, $l = 4$) r-systems comprised of $L_r = 9$ libraries.

into l shards that are written to a cluster of l tapes. These tapes, indicated by the boxes in Figure 8, are mounted to the l drives of a selected drive group. The color of the boxes is the same as that of the corresponding drive group. Subsequently, the additional $m - l$ parity shards are written to $m - l$ parity tapes, indicated by boxes with the same color and the symbol P, to form the corresponding array. The parity tapes reside in $m - l$ of the l libraries of the subsequent tape drive group. Clearly, there are $\binom{l}{m-l}$ different placement combinations. In the case of an MDS(5,4) coding scheme, there are $\binom{4}{5-4} = 4$ combinations. For the red drive group shown in Figure 8(a), these combinations are shown in the first four lines. The red parity tapes reside in the 4 libraries of the subsequent (green) drive group, namely, in the set $\{L_5, L_6, L_7, L_8\}$ of libraries. For the green drive group, these combinations are shown in lines 5-8. The green parity tapes reside in the 4 libraries of the subsequent (blue) drive group, namely, in the set $\{L_9, L_1, L_2, L_3\}$ of libraries. Similarly the blue parity tapes reside in the 4 libraries of the subsequent drive group, namely, in the set $\{L_4, L_5, L_6, L_7\}$. The placement of the user and parity data in the case of an MDS(6,4) coding scheme is shown in Figure 8(b). In this case there are $\binom{4}{6-4} = 6$ combinations for selecting two parity tapes in four libraries. Their placement for the red and green drive groups is shown in the figure. The placement scheme presented ensures that user data and parity tapes are distributed across libraries such that within each library a fraction l/m of its tapes contains user data and the remaining tapes contain parities.

REFERENCES

[1] Tape Roadmap, Information Storage Industry Consortium (INSIC) Report, 2019. [Online]. Available: https://www.insic.org/wp-content/uploads/2019/07/INSIC-Applications-and-Systems-Roadmap.pdf [retrieved: April 2025]

[2] M. A. Lantz, S. Furrer, M. Petermann, H. Rothuizen, S. Brach, L. Kronig, I. Iliadis, B. Weiss, E. R. Childers, and D. Pease, "Magnetic tape storage technology," ACM Trans. Storage, vol. 21, no. 1, Jan. 2025, pp. 1–70.

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

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

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

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

[7] ——, "Relations between entity sizes and error-correction coding codewords and data loss," in Proc. 17th Int'l Conference on Communication Theory, Reliability, and Quality of Service (CTRQ), May 2024, pp. 1–11.

[8] ——, "Relations between entity sizes and error-correction coding codewords and effective data loss," Int'l J. Adv. Networks and Services, vol. 17, no. 3&4, Dec. 2024, pp. 69–94.

[9] I. Iliadis and M. Lantz, "Reliability evaluation of automated tape library systems," in Proc. 32nd IEEE Int'l Symp. on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS), Oct. 2024, pp. 80–87.

[10] D. A. Ford, R. J. T. Morris, and A. E. Bell, "Redundant arrays of independent libraries (RAIL): the StarFish tertiary storage system," Parallel Comput., vol. 24, no. 1, Jan. 1998, p. 45–64.

[11] HPSS: High Performance Storage System, HPSS RAIT TECHNOLOGY, Software technology for striping data on tape with parity for redundancy. [Online]. Available: https://hpss-collaboration.org/rait/ [retrieved: April 2025]

[12] PoINT Archival Gateway Unified Object Storage with Disk and Tape. [Online]. Available: https://www.point.de/fileadmin/user_upload/datenblaetter/2024_point-archival-gateway_unified_technical-white-paper_e_20240216.pdf (2024)

[13] D. S. Papailiopoulos and A. G. Dimakis, "Locally repairable codes," in Proc. 2012 IEEE Int'l Symposium on Information Theory, Jul. 2012, pp. 2771–2775.

[14] IBM TS4300 Tape Library, Overview, Drive Features, Recommended access order (RAO) open function. [Online]. Available: https://www.ibm.com/docs/en/ts4300-tape-library [retrieved: April 2025]

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

[16] Ultrium LTO-8. [Online]. Available: https://www.lto.org/lto-8/ [retrieved: January 2025]

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

[18] ——, "Performance evaluation of tape library systems," Perform. Eval., vol. 157-158, Oct. 2022, pp. 1–22.

[19] IBM TS4500 Tape Library, Systems Hardware Data Sheet. [Online]. Available: https://www.ibm.com/products/ts4500 [retrieved: April 2025]