Babel Storage: Uncoordinated Content Delivery from Multiple Coded Storage Systems

Babel Storage: Uncoordinated Content Delivery from Multiple Coded Storage Systems

Joachim Neu
Stanford University
Stanford, CA, USA
Email: jneu@stanford.edu

Muriel Médard
Massachusetts Institute of Technology
Cambridge, MA, USA
Email: medard@mit.edu

***Abstract*—In future content-centric networks, content is identified independently of its location. From an end-user's perspective, individual storage systems dissolve into a seemingly omnipresent structureless 'storage fog'. Content should be delivered oblivious of the network topology, using multiple storage systems simultaneously, and at minimal coordination overhead. Prior works have addressed the advantages of error correction coding for distributed storage and content delivery separately. This work takes a comprehensive approach to highlighting the tradeoff between storage overhead and transmission overhead in uncoordinated content delivery from multiple coded storage systems.**

Our contribution is twofold. First, we characterize the tradeoff between storage and transmission overhead when all participating storage systems employ the same code. Second, we show that the resulting stark inefficiencies can be avoided when storage systems use diverse codes. What is more, such code diversity is not just technically desirable, but presumably will be the reality in the increasingly heterogeneous networks of the future. To this end, we show that a mix of Reed-Solomon, low-density parity-check and random linear network codes achieves close-to-optimal performance at minimal coordination and operational overhead.

***Index Terms*—Error correction coding, content-centric networks, distributed storage, content delivery.**

I. Introduction

In current communication networks, content is addressed by its (alleged) location, e.g., through its URL. Content delivery is carried out only by the hosting storage system. Different 'hacks', e.g., GeoDNS and IPv4 anycast [1], are used to overcome the shortcomings of this approach. In future *content-centric networks*, content is identified independent of its location [2]. From an end-user's perspective, individual storage systems dissolve into a seemingly omnipresent, structureless and thus flexible 'storage fog'. This storage facility delivers the desired data without requiring the user to know about the data's location. File delivery can be carried out by multiple storage systems simultaneously, and transparently to the user.

Storage systems use error correcting codes [3] to improve reliability in the presence of temporary and permanent disk failures. Different code families have been investigated for application in coded storage systems, such as Reed-Solomon codes (RS-Cs) [4]–[6], random linear network codes (RLN-Cs) [7], low-density parity-check codes (LDPC-Cs) [8]–[11] and locally repairable codes (LR-Cs) [12], [13], among many other codes [14]–[18]. As there is no clear 'best' coding system, a diverse mix of codes is expected to be found among the different storage units of a storage fog. A separate line of works has studied the benefits of introducing redundant information via error correcting codes for content delivery in wireless networks [19], such as for caching [20], [21], using device-to-device communications [22], or in gossip protocols [23], [24].

How can multiple storage systems, employing a mix of different error correcting codes, jointly deliver content to a user with high throughput, low coordination overhead, and low complexity? *In particular, how can we combine the two so far isolated approaches of coding for storage and coding for content delivery?* This is the question guiding our work in this paper. We present our results as follows. In Section II, we introduce the system model. We then look at two extreme cases regarding the used codes. First, in Section III, we use exactly the same code on all participating storage systems. We show that storage redundancy, while originally introduced to improve reliability, also helps in efficient file delivery. We characterize the tradeoff between storage and transmission overhead. Second, in Section IV, we examine the opposite extreme of using different codes. In particular, we combine RS-Cs, RLN-Cs and LDPC-Cs, and demonstrate that the combination behaves similar to a low-field-size random code. We summarize the implications of our findings as relevant to system implementers and network operators in Section V.

II. System Model

We denote the finite field of characteristic p as $\mathbb{F}_p$ and its extension field of degree m as $\mathbb{F}_{p^m}$. We sometimes write $\mathbb{F}_q$ for $\mathbb{F}_{p^m}$ implying that $q = p^m$. Note that $\mathbb{F}_{q^m}$ is isomorphic to the m-dimensional $\mathbb{F}_q$-vector space $\mathbb{F}_q^m$. We use lower-case Greek letters for field elements, lower-case Latin bold letters for vectors, and upper-case Latin bold letters for matrices. As customary in coding theory, vectors denote row vectors unless stated otherwise. Transposition is $(.)^\top$.

Fix a block length n, dimension $k \leq n$, and field $\mathbb{F}_q$, then a $[n,k]_q$ linear block code $\mathcal{C}$ with rate $R \triangleq \frac{k}{n}$ is a k-dimensional subspace of $\mathbb{F}_q^n$. The code can equivalently be represented using a generator matrix $\mathbf{G} \in \mathbb{F}_q^{k\times n}$ or a parity-check matrix $\mathbf{H} \in \mathbb{F}_q^{(n-k)\times n}$,

$$\mathcal{C} = \left\{\mathbf{c} \;\middle|\; \exists\, \mathbf{u} \in \mathbb{F}_q^k : \mathbf{c} = \mathbf{u}\mathbf{G}\right\} = \left\{\mathbf{c} \;\middle|\; \mathbf{H}\mathbf{c}^\top = \mathbf{0}^\top\right\} \subseteq \mathbb{F}_q^n. \quad (1)$$

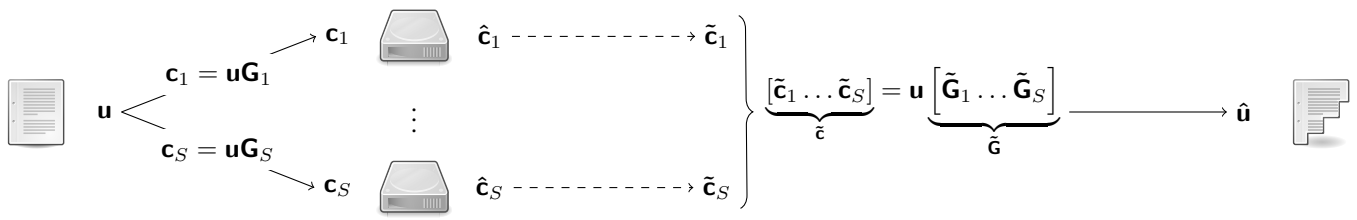

Fig. 1. A file is stored and distributed using S different storage systems in the following way. For each storage system $i \in \{1, ..., S\}$, the file $\mathbf{u}$ is encoded into $\mathbf{c}_i$ which is stored. Some symbols of $\mathbf{c}_i$ get erased over time; the remaining ones are collected in the vector $\hat{\mathbf{c}}_i$. Of these, some are chosen for transmission; the symbols that arrive at the requesting user are denoted as $\tilde{\mathbf{c}}_i$. As long as $\operatorname{rank}\left[\tilde{\mathbf{G}}_1 \dots \tilde{\mathbf{G}}_S\right] = k$, the file $\hat{\mathbf{u}} = \mathbf{u}$ can be decoded.

Note that neither $\mathbf{G}$ nor $\mathbf{H}$ are unique and that $\mathbf{G}$ fixes the mapping between information symbols $\mathbf{u}$ and codeword symbols $\mathbf{c}$ as

$$\mathbf{c} = \mathbf{u}\mathbf{G}. \tag{2}$$

During storage and transmission, some number n_{E} of symbols of the codeword $\mathbf{c}$ might get erased, so that only a subset $\tilde{\mathbf{c}} \in \mathbb{F}_q^{n-n_{\mathrm{E}}}$ is received. The $\tilde{\mathbf{c}}$ relate to $\mathbf{u}$ as $\tilde{\mathbf{c}} = \mathbf{u}\tilde{\mathbf{G}}$ via a reduced $\tilde{\mathbf{G}} \in \mathbb{F}_q^{k\times(n-n_{\mathrm{E}})}$ where those columns of $\mathbf{G}$ are removed that correspond to the erased symbols in $\mathbf{c}$. As long as $\tilde{\mathbf{G}}$ is full rank (i.e., $\operatorname{rank}\tilde{\mathbf{G}} = k$), $\tilde{\mathbf{c}} = \mathbf{u}\tilde{\mathbf{G}}$ has a unique solution and $\mathbf{u}$ can be recovered from $\tilde{\mathbf{c}}$, e.g., using Gaussian elimination in runtime complexity $\mathcal{O}(k^3)$.

The storage and distribution of a file using S different storage systems is depicted in Fig. 1 and proceeds as follows. For each storage system $i \in \{1, ..., S\}$, the file $\mathbf{u} \in \mathbb{F}_q^k$ is encoded using a $[n_i, k]_q$ code into $\mathbf{c}_i = \mathbf{u}\mathbf{G}_i \in \mathbb{F}_q^{n_i}$. These codewords $\mathbf{c}_i$ are stored. Some symbols get erased over time, the remaining ones are denoted $\hat{\mathbf{c}}_i \in \mathbb{F}_q^{\hat{n}_i}$. Of these, some are chosen for transmission, so that $\tilde{\mathbf{c}}_i \in \mathbb{F}_q^{\tilde{n}_i}$ arrive at the requesting user. Let $\tilde{\mathbf{c}} \triangleq [\tilde{\mathbf{c}}_1 \dots \tilde{\mathbf{c}}_S]$ and $\tilde{\mathbf{G}} \triangleq \left[\tilde{\mathbf{G}}_1 \dots \tilde{\mathbf{G}}_S\right]$. As long as $\operatorname{rank}\tilde{\mathbf{G}} = k$, the file $\hat{\mathbf{u}} = \mathbf{u}$ can be decoded.

The goal is to transmit as few symbols as possible, but enough so that the user can decode as soon as possible with very high probability. A codeword symbol is useful if and only if its corresponding column increases the rank of the matrix $\tilde{\mathbf{G}}$. There should be no coordination, neither among the storage systems nor by the user, regarding the selection of symbols to be transmitted. Instead, using the metaphor of a 'digital fountain' [25], upon request each storage system sends a stream of symbols until the user asks it to stop. The system performance is dominated by a *coupon collector's problem* – how many transmissions (some of which might not add new information) are necessary to collect the required k independent pieces? In Sections III and IV we show how storage redundancy and code diversity can mitigate the loss in efficiency due to inadvertently transmitting duplicate pieces.

III. Minimum Code Diversity

In this section, we consider an extreme case of the system depicted in Fig. 1. All S storage systems use the same $[n, k]_q$ maximum distance separable (MDS) code, i.e., the information can be decoded from any k codeword symbols (and equivalently, any k columns of the code's $\mathbf{G}$ are linearly independent). The storage systems take turns transmitting a randomly selected codeword symbol to the user, e.g., for $S = 2$, the 1st, 3rd, ..., and 2nd, 4th, ..., symbols arrive from the first and second system, respectively. While each storage system ensures uniqueness among the symbols it sends, the user might receive the same symbol (and thus redundant information) multiple times, from different systems.

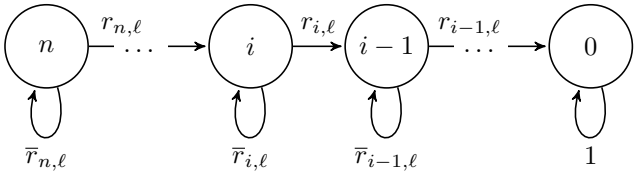

Fig. 2. Time-varying Markov chain modeling the evolution of the number of symbols U_ℓ that were not seen among the first ℓ receive symbols.

In the following, we analyze the underlying coupon collector's problem, both for large systems (block length $n \to \infty$) as well as for systems with finite n. We model the number of symbols that were not seen among the first ℓ received symbols as the random variables U_ℓ ('unknown'). The $\{U_\ell\}_{\ell=0}^{S_n}$ are described by the time-varying Markov chain depicted in Fig. 2, with states $\mathcal{U} = \{0, ..., n\}$, initial state U_0 distributed as

$$U_0 = \begin{cases} n & \text{with probability (w.p.) } 1 \end{cases} \tag{3}$$

and transitions

$$U_{\ell+1} \mid U_\ell = \begin{cases} U_\ell - 1 & \text{w.p. } r_{U_\ell,\ell} \\ U_\ell & \text{w.p. } \overline{r}_{U_\ell,\ell} \end{cases} \tag{4}$$

where

$$r_{i,\ell} \triangleq \frac{i}{n - \left\lfloor \frac{\ell}{S} \right\rfloor} \qquad \text{and} \qquad \overline{r}_{i,\ell} \triangleq 1 - r_{i,\ell} \tag{5}$$

model coupon collecting and denote the probability of receiving a new or duplicate symbol in the ℓ-th transmission, respectively, if at that time i symbols are unknown to the receiver. In the ℓ-th round, the active storage system chooses uniformly at random one of the $n - \left\lfloor \frac{\ell}{S} \right\rfloor$ pieces it has not

transmitted in previous rounds. Out of these pieces, i pieces have not been seen previously by the receiver, hence the probability $r_{i,\ell}$ for receiving novel information. Using this model, the probability mass functions (PMFs) $P_{U_\ell}(u_\ell)$ can be obtained for all ℓ for any fixed n and S.

Lemma 1. *The following recursion holds:*

$$\mathbb{E}[U_{\ell+1}] = \underbrace{\left(1 - \frac{1}{n - \left\lfloor \frac{\ell}{S} \right\rfloor}\right)}_{\triangleq a_\ell} \mathbb{E}[U_\ell] \tag{6}$$

Proof.

$$\mathbb{E}[U_{\ell+1}] = \mathbb{E}[\mathbb{E}[U_{\ell+1}|U_\ell]] \tag{7}$$

$$= \mathbb{E}[(U_\ell - 1)\, r_{U_\ell,\ell} + U_\ell\, \overline{r}_{U_\ell,\ell}] = a_\ell\, \mathbb{E}[U_\ell] \tag{8}$$

□

Corollary 1.

$$\mathbb{E}[U_\ell] = n \prod_{i=0}^{\ell-1} \left(1 - \frac{1}{n - \left\lfloor \frac{i}{S} \right\rfloor}\right) = n \prod_{i=0}^{\ell-1} a_i \tag{9}$$

For ease of exposition, we introduce random variables for the number of received unique symbols, $K_\ell \triangleq n - U_\ell$, and rescaled variants with normalized time and amount of data,

$$U_\tau^{(n)} \triangleq \frac{1}{n} U_{n\tau}, \qquad K_\tau^{(n)} \triangleq \frac{1}{n} K_{n\tau}, \qquad \tau \in [0, S]. \tag{10}$$

The following theorem provides an analytic expression for the system behavior in the large block length regime ($n \to \infty$):

Theorem 1. *For $n \to \infty$, the fraction of unseen symbols at any given point in time τ concentrates to its mean and follows*

$$u_S(\tau) \triangleq \lim_{n\to\infty} \mathbb{E}\left[U_\tau^{(n)}\right] = \left(1 - \frac{\tau}{S}\right)^S. \tag{11}$$

Proof. First, we show the limit of the expectation.

$$\lim_{n\to\infty} \log\left(\mathbb{E}\left[U_\tau^{(n)}\right]\right) = \lim_{n\to\infty} \sum_{\ell=0}^{n\tau-1} \log\left(1 - \frac{1}{n - \left\lfloor \frac{\ell}{S} \right\rfloor}\right) \tag{12}$$

$$\overset{(a)}{\approx} \lim_{n\to\infty} \sum_{\ell=0}^{n\tau-1} \log\left(1 - \frac{1}{n - \frac{\ell}{S}}\right) \overset{(b)}{\approx} \lim_{n\to\infty} -\sum_{\ell=0}^{n\tau-1} \frac{1}{n - \frac{\ell}{S}} \tag{13}$$

$$= \lim_{n\to\infty} -\sum_{\ell=0}^{n\tau-1} \frac{1}{n} \frac{1}{1 - \frac{\ell}{Sn}} \overset{(c)}{\approx} \int_0^\tau \frac{1}{\frac{x}{S} - 1}\, \mathrm{d}x \tag{14}$$

$$= S \log\left(1 - \frac{\tau}{S}\right) \tag{15}$$

Here, (a) uses $\left\lfloor \frac{\ell}{S} \right\rfloor \approx \frac{\ell}{S}$ for large ℓ, (b) uses $\log(1-x) \approx -x$ for $x \approx 0$, and (c) uses the convergence of the left Riemann sum to the Riemann integral,

$$\lim_{n\to\infty} \sum_{i=0}^{\alpha n-1} \frac{1}{n} f\left(\frac{i}{n}\right) = \int_0^\alpha f(x)\, \mathrm{d}x, \tag{16}$$

for $\alpha \triangleq \tau$, $f(x) \triangleq \frac{1}{\frac{x}{S}-1}$, and $S > \tau > 0$.

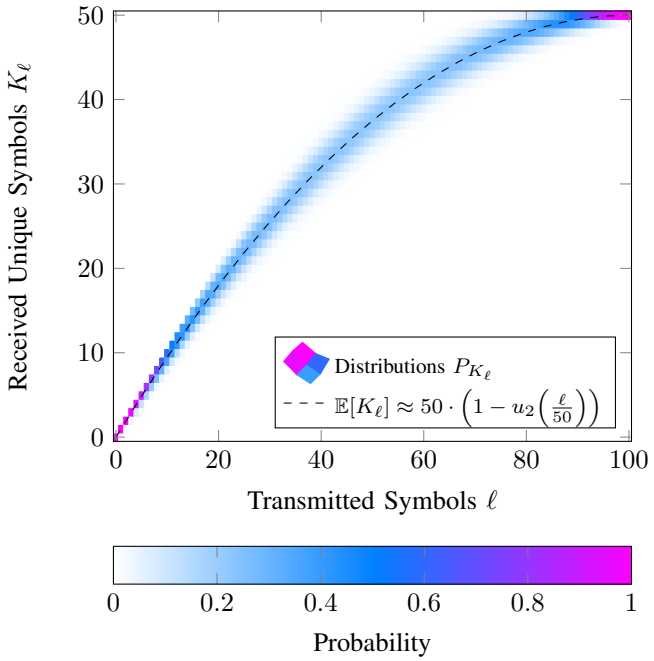

Fig. 3. PMFs $P_{K_\ell}(k_\ell)$ of the number of received unique symbols K_ℓ after transmitting ℓ symbols, for $S = 2$ storage systems and block length $n = 50$, and approximation $\mathbb{E}[K_\ell] \approx n \cdot \left(1 - u_S\left(\frac{\ell}{n}\right)\right)$ according to Theorem 1.

Second, we show the concentration of $U_\tau^{(n)}$ to its expectation as $n \to \infty$. Note that

$$\forall S, n: \quad \mathbb{E}[U_\ell] \le n \left(1 - \frac{\ell}{Sn}\right)^S \le n \exp\left(-\frac{\ell}{n}\right), \tag{17}$$

$$\forall \ell \ \ \forall \ell' \le \ell: \quad 0 \le a_\ell \le a_{\ell'} \le 1, \qquad 2a_\ell - 1 \le a_\ell^2. \tag{18}$$

Hence,

$$\mathrm{Var}[U_{\ell+1}] \overset{(a)}{=} (2a_\ell - 1)\mathbb{E}\left[U_\ell^2\right] - a_\ell^2 \mathbb{E}[U_\ell]^2 + (1 - a_\ell)\mathbb{E}[U_\ell] \tag{19}$$

$$\overset{(b)}{\le} (1 - a_\ell)\mathbb{E}[U_\ell] + a_\ell^2\, \mathrm{Var}[U_\ell] \tag{20}$$

$$\overset{(c)}{\le} \sum_{i=0}^{\ell} (1 - a_{\ell-i})\mathbb{E}[U_{\ell-i}] \prod_{j=0}^{i-1} a_{\ell-j}^2 \tag{21}$$

$$\overset{(d)}{\le} n \sum_{i=0}^{\ell} (1 - a_{\ell-i}) \exp\left(-\frac{\ell - i}{n}\right) \tag{22}$$

$$\overset{(e)}{\le} \frac{n}{n - \left\lfloor \frac{\ell}{S} \right\rfloor} \sum_{i=0}^{\ell} \exp\left(-\frac{i}{n}\right) = \mathcal{O}(n). \tag{23}$$

Here, (a) uses Lemma 1 and $\mathrm{Var}[X] = \mathbb{E}\left[X^2\right] - \mathbb{E}[X]^2$, (b) uses (18), (c) results from repeated application of inequality (20), (d) uses (17) and (18), and (e) uses (18). As a result, using Chebyshev's inequality, for any fixed $\varepsilon > 0$,

$$\lim_{n\to\infty} \Pr\left[\left| U_\tau^{(n)} - \mathbb{E}\left[U_\tau^{(n)}\right] \right| \ge \varepsilon\right] \le \lim_{n\to\infty} \frac{\mathrm{Var}[U_{n\tau}]}{\varepsilon^2 n^2} = 0. \tag{24}$$

□

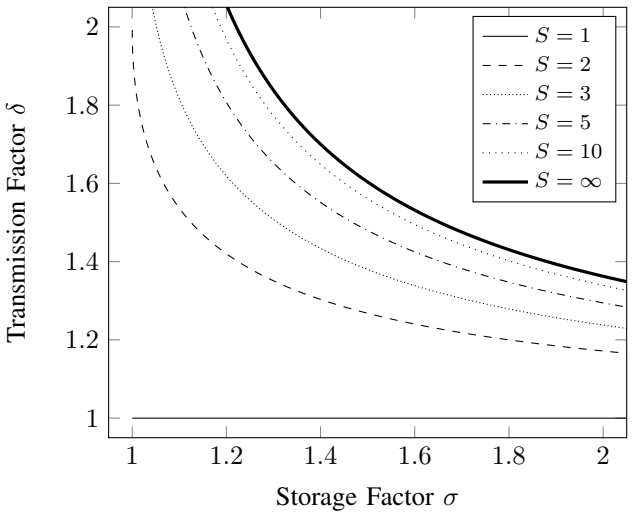

Fig. 4. Tradeoff between storage factor σ and transmission factor δ for different numbers of storage systems S at large block length $n \to \infty$.

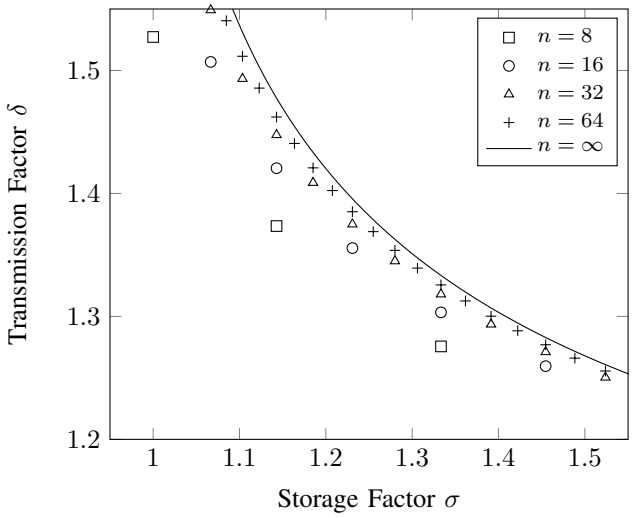

Fig. 5. Tradeoff between storage factor σ and transmission factor δ for number of storage systems $S = 2$ and different block lengths n. Already for reasonably low $n \approx 64$ the system behavior is well predicted using the analytic expression for the asymptotic regime $n \to \infty$.

Ergo, as $n \to \infty$, the fraction of unique received symbols is fully characterized by $1 - u_S(\tau) = 1 - \left(1 - \frac{\tau}{S}\right)^S$. Fig. 3 visualizes the PMFs $P_{K_\ell}(k_\ell)$ of the number of received unique symbols K_ℓ after transmitting ℓ symbols, for the example of $S = 2$ and $n = 50$. The approximation based on Theorem 1, $\mathbb{E}[K_\ell] \approx n \cdot \left(1 - u_S\left(\frac{\ell}{n}\right)\right)$, fits well.

To analyze the system for finite block lengths, we turn to the random variable L denoting the number of transmissions required to complete the download (i.e., to receive k unique symbols). Exactly ℓ transmissions are required if the user receives $k-1$ unique symbols in the first $\ell - 1$ transmissions and a new symbol in the ℓ-th transmission, thus,

$$P_L(\ell) = P_{U_{\ell-1}}(n - k + 1) \cdot r_{n-k+1,\ell-1}. \tag{25}$$

The average number of transmissions required to complete the download is then obtained as the expected value $\tilde{L} \triangleq \mathbb{E}[L]$. We normalize this to $\tilde{\tau} \triangleq \frac{\tilde{L}}{n}$. In a similar fashion for $n \to \infty$, $\tilde{\tau}$ is the fraction of time (i.e., number of transmissions) it takes to complete the download, which can be obtained from

$$R \stackrel{!}{=} 1 - \left(1 - \frac{\tilde{\tau}}{S}\right)^S, \qquad \tilde{\tau} = S \, \left(1 - \sqrt[S]{1 - R}\right). \tag{26}$$

Assume the storage systems use an MDS code of rate $R = \frac{k}{n}$. We call $\sigma \triangleq \frac{1}{R} = \frac{n}{k}$ the *storage factor* quantifying the storage overhead of the systems due to coding. Similarly, we call $\delta \triangleq \frac{\tilde{\tau}}{R}$ the *transmission factor*, which quantifies the overhead in transmissions required for uncoordinated vs. coordinated (i.e., where duplicate symbols are avoided) download.

Fig. 4 shows the tradeoff between σ and δ for large n and different number of storage systems S all employing the same MDS code. Clearly, uncoordinated file delivery incurs a loss in transmission efficiency. The loss gets worse the more independent servers participate in the delivery. However, even for an arbitrarily large number of sources, the loss remains bounded as long as $\sigma \gtrsim 1$. The storage redundancy can be used to mitigate the transmission overhead.

Fig. 5 shows the behavior for some small block lengths n and two servers $S = 2$. In particular for σ close to 1, the system completes downloads slightly faster than for $n \to \infty$. Note, however, that even for moderate code lengths $n \approx 64$ the asymptotic expression is a good proxy for the system behavior.

IV. Maximum Code Diversity

After examining the extreme case of minimum code diversity in the previous section, we turn to the opposite extreme, maximum code diversity, in this section. The possibilities of combining codes of different families have previously been investigated, for instance in [26]. We explore the interoperability of three systems employing an RLN-C, RS-C and LDPC-C, respectively, in an experimental case study of the framework depicted in Fig. 1. We assume the RLN-C and RS-C are of parameters $[160, 128]_{256}$ and constructed in the usual way [27]–[29]. As LDPC-C we use the AR4JA code [30], [31] of equivalent parameters $[1280, 1024]_2$.

Linear codes can be defined as the nullspace of their parity-check matrices $\mathbf{H}$. To combine the codes, we transform the defining systems of linear equations to the common base field $\mathbb{F}_2$ as in [26], [32]. Let $\{\mathbf{X}\}_y$ denote the 'y-th' entry of $\mathbf{X}$. The parity-check equations in $\mathbb{F}_{p^m}$,

$$\mathbf{H}\mathbf{c}^\top = \mathbf{0}^\top \iff \forall i \in \{1, ..., n-k\} : \sum_{j=1}^{n} \{\mathbf{H}\}_{ij} \{\mathbf{c}\}_j = 0, \tag{27}$$

can be equivalently expressed in $\mathbb{F}_p$ (albeit with a larger system of equations), once $+$ and $\cdot$ in $\mathbb{F}_{p^m}$ are reduced to operations

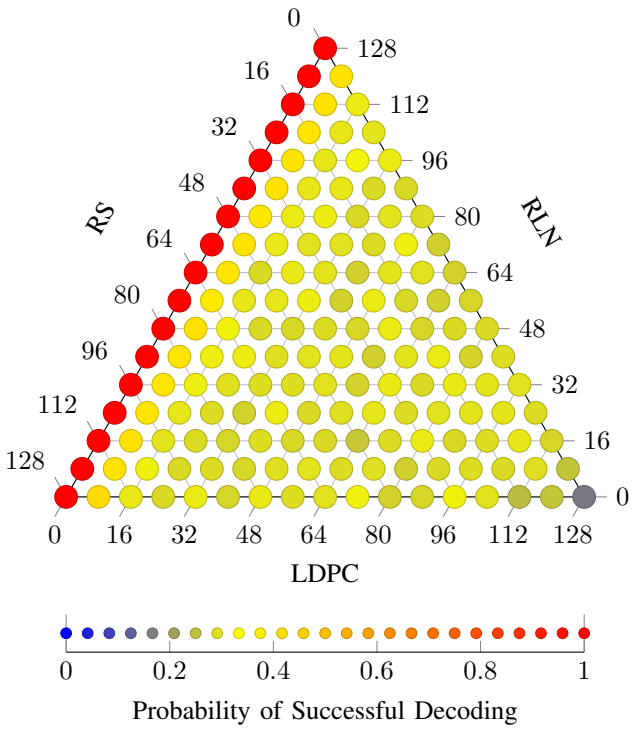

Fig. 6. Probability of successful decoding (mark color) after downloading the minimum number of $k = 128$ symbols, consisting of a mix of RLN, RS and LDPC coded symbols (coordinate). The upper center, lower left, and lower right vertex correspond to downloading only RLN, only RS, and only LDPC coded symbols, respectively. Other points correspond to interpolated mixes.

in $\mathbb{F}_p^m$. Let $\mathfrak{v} : \mathbb{F}_{p^m} \to \mathbb{F}_p^m$ be the isomorphism between $\mathbb{F}_{p^m}$ and $\mathbb{F}_p^m$, and $\mathfrak{v}^{-1} : \mathbb{F}_p^m \to \mathbb{F}_{p^m}$ its inverse. Clearly,

$$\forall \alpha, \beta \in \mathbb{F}_{p^m}, \gamma \in \mathbb{F}_p : \ \mathfrak{v}\left(\alpha + \gamma \cdot \beta\right) = \mathfrak{v}\left(\alpha\right) + \gamma \cdot \mathfrak{v}\left(\beta\right). \quad (28)$$

Let $f_\alpha(\beta) \triangleq \alpha \cdot \beta$. Since $+$ and $\cdot$ are distributive, $\tilde{f}_\alpha\left(\mathbf{b}\right) \triangleq \mathfrak{v}\left(f_\alpha\left(\mathfrak{v}^{-1}\left(\mathbf{b}\right)\right)\right)$ is linear, and thus equivalently represented by a matrix $\mathfrak{m}\left(\alpha\right) \in \mathbb{F}_p^{m \times m}$ associated with α,

$$\tilde{f}_\alpha\left(\mathbf{b}\right) = \mathfrak{m}\left(\alpha\right)\mathbf{b}. \quad (29)$$

We thus get the equivalent system of equations in $\mathbb{F}_p$,

$$\forall i \in \{1, ..., n-k\} : \ \sum_{j=1}^{n} \mathfrak{m}\left(\left\{\mathbf{H}\right\}_{ij}\right) \mathfrak{v}\left(\left\{\mathbf{c}\right\}_j\right) = \mathbf{0}. \quad (30)$$

The construction applies analogously to the codes' generator matrices $\mathbf{G}$. While technically all three codes have parameters $[1280, 1024]_2$ after this 'lifting operation' from $\mathbb{F}_{256}$ to $\mathbb{F}_2$, we do not pick columns from the lifted generator matrices independently. Rather, we adopt block-aligned erasures, where blocks of eight columns are either jointly erased or jointly present, mimicking the erasure of $\mathbb{F}_{256}$ rather than $\mathbb{F}_2$ symbols. For the RLN-C and RS-C, this preserves their properties (both codes behave considerably worse under random erasure of $\mathbb{F}_2$ sub-symbols), while the LDPC-C's performance is unaffected.

Fig. 6 shows the probability of the user being able to decode (i.e., the matrix $\tilde{\mathbf{G}}$ being full-rank) after downloading the theoretical minimum of $k' = 8 \cdot 128 = 1024$ base field $\mathbb{F}_2$ symbols in a block-aligned fashion to mimic $\mathbb{F}_{256}$ symbols. It can be seen that RS-C and RLN-C symbols can be mixed without performance penalty and the close-to-optimal performance of RLN-Cs over large fields is preserved. As soon as LDPC-C symbols enter the mix, downloading the theoretical minimum of symbols is likely not sufficient. However, after downloading one, two and three additional $\mathbb{F}_{256}$ symbols, the probability of successful decoding reaches 90%, 99% and 99.9%, respectively, independent of the mixture – which is 2.34% transmission overhead (for 131 instead of 128 symbol transmissions). This is the performance one would expect for an RLN-C over small fields. In fact, when replacing the LDPC-C with a binary RLN-C, the qualitative system behavior remains unchanged.

Note that erasure decoding using Gaussian elimination over $\tilde{\mathbf{G}}$ was performed here. Its cubic runtime complexity is acceptable in the case at hand, as the resulting systems of linear equations are of manageable size, and the cost of Gaussian elimination is amortized over the length of the packets. Belief propagation decoding [33] is not feasible as neither RLN-Cs nor RS-Cs have the required sparse structure.

As a result of our empirical case study, we conclude that for system design purposes in uncoordinated content delivery, binary LDPC-Cs can be treated as binary RLN-Cs. The system incurs a small degradation in transmission efficiency due to the small field size, but the overall performance stays close to the theoretical optimum. Decoding is possible with high probability after downloading only $k + \Delta k$ symbols, with a small Δk (in the order of three) number of excess symbols.

In contrast to the degradation observed in Section III from using the same code across multiple storage systems, code diversity enables close-to-optimal uncoordinated content delivery. Note, that lifting to the base field and random recoding transforms any code into an RLN-C over $\mathbb{F}_2$, providing an effective means to increase code diversity in existing systems.

V. Conclusion

In this paper we analyzed the performance of uncoordinated file delivery from multiple coded storage systems. System implementers and network operators should draw the following conclusions and actionable recommendations from our results. In Section III, we showed how uncoordinated content delivery from multiple storage systems using an identical code incurs a loss in transmission efficiency (in comparison to coordinated content delivery) due to a coupon collector effect. In Section IV, we showed that this degradation can be resolved (almost perfectly) by using different linearly independent codes in all the storage systems.

To leverage the benefits of coding for efficient uncoordinated content delivery from multiple sources, implementers should make the raw storage-coded data and details about the employed coding techniques available to the transport layer, rather than treating storage-coding as a hidden internal of the storage system. Implementers and operators should employ code families that allow the design of larger numbers of

linearly independent codes. RLN-Cs and RS-Cs over larger field sizes are natural candidates. Network operators should assign codes to storage systems carefully to maintain proper code diversity (analogous to frequency reuse in mobile network base stations). When the same code is used multiple times, random remixing/recoding of the stored symbols before transmission can be used to emulate (close-to-optimal) RLN-C performance. Finally, the procedure outlined in this paper lets system operators change the employed codes without requiring downtime or instantaneous recoding of the whole database.

REFERENCES

[1] P. L. Dordal, *An Introduction to Computer Networks*, 2018, release 1.9.16. [Online]. Available: http://intronetworks.cs.luc.edu/

[2] T. Koponen *et al.*, "A data-oriented (and beyond) network architecture," in *Proc. ACM SIGCOMM Conf. Appl., Technol., Architectures, and Protocols for Comput. Commun.*, 2007, pp. 181–192.

[3] M. Bossert, *Channel Coding for Telecommunications*. Wiley, 1999.

[4] A. V. Goldberg and P. N. Yianilos, "Towards an archival intermemory," in *Proc. IEEE Forum Res. and Technol. Advances in Digit. Libraries (ADL)*, 1998, pp. 147–156.

[5] J. Kubiatowicz *et al.*, "OceanStore: An architecture for global-scale persistent storage," in *Proc. Int. Conf. Architectural Support for Program. Lang. and Operating Syst. (ASPLOS)*, 2000, pp. 190–201.

[6] K. V. Rashmi *et al.*, "A solution to the network challenges of data recovery in erasure-coded distributed storage systems: A study on the Facebook warehouse cluster," in *Proc. USENIX Workshop Hot Topics in Storage and File Syst. (HotStorage)*, 2013.

[7] S. Acedanski *et al.*, "How good is random linear coding based distributed networked storage?" in *Workshop Network Coding, Theory and Applicat. (NETCOD)*, Riva del Garda, Italy, 2005.

[8] J. S. Plank and M. G. Thomason, "On the practical use of LDPC erasure codes for distributed storage applications," University of Tennessee, Tech. Rep. CS-03-510, September 2003.

[9] H. Park, D. Lee, and J. Moon, "LDPC code design for distributed storage: Balancing repair bandwidth, reliability, and storage overhead," *IEEE Trans. Commun.*, vol. 66, no. 2, pp. 507–520, 2018.

[10] M. G. Luby *et al.*, "Liquid cloud storage," *arXiv:1705.07983v1*, 2017.

[11] M. Luby, "Capacity bounds for distributed storage," *arXiv:1610.03541v5*, 2018.

[12] M. Sathiamoorthy *et al.*, "XORing elephants: Novel erasure codes for big data," *Proc. VLDB Endowment*, vol. 6, no. 5, pp. 325–336, 2013.

[13] D. S. Papailiopoulos and A. G. Dimakis, "Locally repairable codes," *IEEE Trans. Inf. Theory*, vol. 60, no. 10, pp. 5843–5855, 2014.

[14] C. Huang *et al.*, "Erasure coding in Windows Azure storage," in *Proc. USENIX Annu. Tech. Conf.*, 2012, pp. 15–26.

[15] A. G. Dimakis *et al.*, "A survey on network codes for distributed storage," *Proc. IEEE*, vol. 99, no. 3, pp. 476–489, 2011.

[16] M. A. Shokrollahi and M. Luby, "Raptor codes," *Foundations and Trends in Commun. and Inf. Theory*, vol. 6, no. 3-4, pp. 213–322, 2009.

[17] M. Li and P. P. C. Lee, "STAIR codes: A general family of erasure codes for tolerating device and sector failures," *ACM Trans. Storage (TOS)*, vol. 10, no. 4, pp. 14:1–14:30, Oct. 2014.

[18] M. Cunche and V. Roca, "Optimizing the error recovery capabilities of LDPC-staircase codes featuring a Gaussian elimination decoding scheme," in *Proc. IEEE Int. Workshop Signal Process. for Space Commun. (SPSC)*, 2008, pp. 1–7.

[19] G. Liva, E. Paolini, and M. Chiani, "Performance versus overhead for fountain codes over F_q," *IEEE Commun. Lett.*, vol. 14, no. 2, pp. 178–180, 2010.

[20] M. A. Maddah-Ali and U. Niesen, "Cache-aided interference channels," *IEEE Trans. Inf. Theory*, vol. 65, no. 3, pp. 1714–1724, 2019.

[21] H. Reisizadeh, M. A. Maddah-Ali, and S. Mohajer, "Erasure coding for decentralized coded caching," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, 2018, pp. 1715–1719.

[22] A. Piemontese and A. Graell i Amat, "MDS-coded distributed caching for low delay wireless content delivery," *IEEE Trans. Commun.*, vol. 67, no. 2, pp. 1600–1612, 2019.

[23] S. Deb, M. Médard, and C. Choute, "Algebraic gossip: A network coding approach to optimal multiple rumor mongering," *IEEE Trans. Inf. Theory*, vol. 52, no. 6, pp. 2486–2507, 2006.

[24] B. Haeupler, "Analyzing network coding (gossip) made easy," *J. ACM*, vol. 63, no. 3, pp. 26:1–26:22, 2016.

[25] J. W. Byers *et al.*, "A digital fountain approach to reliable distribution of bulk data," in *Proc. ACM SIGCOMM Conf. Appl., Technol., Architectures, and Protocols for Comput. Commun.*, 1998, pp. 56–67.

[26] C. Hellge and M. Médard, "Multi-code distributed storage," in *Proc. IEEE Int. Conf. Cloud Comput. (CLOUD)*, 2016, pp. 839–842.

[27] T. Ho *et al.*, "A random linear network coding approach to multicast," *IEEE Trans. Inf. Theory*, vol. 52, no. 10, pp. 4413–4430, 2006.

[28] I. S. Reed and G. Solomon, "Polynomial codes over certain finite fields," *J. SIAM*, vol. 8, no. 2, pp. 300–304, 1960.

[29] R. R. Borujeny and M. Ardakani, "A new class of rateless codes based on Reed-Solomon codes," *IEEE Trans. Commun.*, vol. 64, no. 1, pp. 49–58, 2016.

[30] D. Divsalar *et al.*, "Protograph based LDPC codes with minimum distance linearly growing with block size," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, 2005, pp. 1152–1156.

[31] "TM synchronization and channel coding – summary of concept and rationale," CCSDS SLS-C&S Working Group, Tech. Rep. 130.1-G-2, November 2012.

[32] J. Blmer *et al.*, "An XOR-based erasure-resilient coding scheme," Int. Comput. Sci. Inst. (ICSI) Berkeley, Tech. Rep. TR-95-048, August 1995.

[33] T. J. Richardson and R. L. Urbanke, *Modern Coding Theory*. Cambridge University Press, 2008.