

RESEARCH

Open Access



Performance and time improvement of LT code-based cloud storage

Nastaran Chakani¹, Seyed Masoud Mirrezaei^{1*}  and Ghosheh Abed Hodtani²

*Correspondence:
sm.mirrezaei@shahroodut.ac.ir

¹ Faculty of Electrical
Engineering, Shahrood University
of Technology, Shahrood, Iran

² Faculty of Electrical
Engineering, Ferdowsi University,
Mashhad, Iran

Abstract

Outsourcing data on cloud storage services has already attracted great attention due to the prospect of rapid data growth and storing efficiencies for customers. The coding-based cloud storage approach can offer a more reliable and faster solution with less storage space in comparison with replication-based cloud storage. LT codes are the famous member of the rateless code family that can improve performance of storage systems utilizing good degree distributions. Since degree distribution plays a key role in LT codes performance, recently introduced Poisson robust soliton distribution (PRSD) and combined Poisson robust soliton distribution (CPRSD) motivate us to investigate LT code-based cloud storage system. So, we exploit LT codes with new degree distributions to provide lower average degree and higher decoding efficiency, specifically when receiving fewer encoding symbols, compared with popular degree distribution, robust soliton distribution (RSD). In this paper, we show that proposed cloud storage outperforms traditional ones in terms of storage space and robustness encountering unavailability of encoding symbols, due to compatible properties of PRSD and CPRSD with cloud storage essence. Furthermore, a modified decoding process is presented based on required encoding symbols behavior to reduce data retrieval time. Numerical results confirm improvement of cloud storage performance.

Keywords: Cloud storage, Degree distributions, LT codes, Retrieval time

1 Introduction

The digital world meets rapid evolutions due to the huge growth of data. According to International Data Corporation (IDC) report, data will grow up to 175 Zettabytes by 2025 [1]. Storing data follows a lot of burdens for owners such as constraints of physical devices, data recovery and also the concern of inaccessibility and cyberattacks. Cloud storage has provided an efficient and promising solution to relieve organizations and different users of data storing issues since the mid-1990s. Cloud storage has several advantages such as the feasibility to access outsourced data anytime from anywhere, through the Internet and to contribute stored data with permission. Fast immigration toward cloud services and close contest of providers to offer the best facilities with reasonable cost are great motivations to find novel solutions to improve cloud storage performance. Generally, cloud storage works based on the fragmentation of data content and distribution of fragments over storage nodes. Two main techniques are employed to ensure data

reliability and availability in distributed storage systems: replication and coding. Replication is the most straightforward method, distributing copies of data over storage nodes. High storage space requirement is the main drawback of replication-based storage, and it is not suitable for distributed systems with high concurrent access requests. Coding-based storage systems can achieve reliability many orders of magnitude higher than replication-based systems for the same redundancy level [2]. There have been plenty of works on erasure codes-based distributed systems [3–7]. High decoding complexity and communication cost to repair a corrupted data fragment reduce popularity of this solution. In order to generate a corrupted encoded packet, usually, reconstruction of all the original packets is required. Thus, the communication cost to repair is equal to the size of the entire original. Reduction in repair communication cost can be achieved by network coding-based storage which works through combining encoded packets in healthy nodes. The utilization of Gaussian elimination decoding in network codes and optimal erasure codes makes them inefficient [8–10]. Near-optimal codes such as LT codes with low complexity encoding and decoding have been proposed for reliable cloud storage systems. If near-optimal codes are exploited, the variation of one original symbol only changes a few numbers of encoded symbols, whereas, in an optimal codes-based system, almost all encoded symbols may be affected by a little bit of modification. Although the number of original packets may be smaller in previous methods, the decoding process performs much faster than in the other storage services [11, 12]. In this paper, we investigate the performance of LT code-based cloud storage with two recent degree distributions in addition to robust soliton distribution. We study the effects of different parameters value on successful data retrieval. In the following, we propose new algorithm to reduce the retrieval time of user data. The rest of this paper is organized as follows. In Sect. 2, related works are reviewed briefly. In Sect. 3, we present LT codes descriptions and their degree distributions. Our system model is stated in Sect. 4. We discuss the performance analysis such as parameter selection, simulation results comparison, and time improvement in Sect. 5. Finally, in Sect. 6, we bring a conclusion to the paper.

2 Related works

Coding has been widely used in cloud storage systems to improve their performance in different aspects. Local reconstruction codes (LRC) as a subset of erasure codes are introduced in [5], keeping storage overhead low compared with Reed–Solomon codes in Windows Azure Storage. Significant decrease in bandwidth and I/Os during repair and improvement in latency for large I/Os are achieved by LRC. In [6], a distributed storage system provides fast content downloads by encoding contents with maximum separable codes (MDS) and applying fork-join queuing for user requests. Results show an essential trade-off between expected download time and storage space which can be useful in the design of a system with delay constraints.

Exploiting LT codes with speculative access mechanisms for parallel writing and reading in a distributed storage architecture leads to high and robust performance [11, 13, 14]. Due to introducing symmetric data redundancy and rateless property of LT codes, the proposed system has high flexibility on data access and improvement compared to traditional parallel storage systems. In [12, 15], a secure cloud storage service is designed

with near-optimal LT codes to solve the reliability issue. The proposed scheme presents efficient data retrieval by exploiting the fast belief propagation decoding algorithm and also utilizes the public integrity verification which helps the data owner to be free from the burden of being online. Proposed scheme presents efficient data retrieval by exploiting the fast belief propagation decoding algorithm and moreover utilizes public integrity verification which helps the data owner be free from the burden of being online. Employing exact repair minimizes data repair complexity and reduces cost. Although the performance analysis and experimental results show an equivalent storage and communication cost in comparison with other erasure codes-based systems, this secure cloud storage service achieves a much faster data retrieval. Compared to network coding-based storage systems, the proposed service reduces storage costs and provides faster data retrieval with comparable communication costs.

In [16], LT-based architecture was proposed for the back-end of block-level cloud (BLC) storage that achieves sufficient levels of performance in terms of access and transfer, availability, integrity, and confidentiality. Interesting features of LT codes such as low complexity and on-the-fly redundancy setting make them suitable for the BLC system. Results indicate that by applying appropriate system parameters, good compromise can be achieved, and the proposed BLC outperforms traditional ones.

The main trade-off between file retrieval delay and successful decoding probability is investigated in a distributed cloud storage system [17]. The proposed multi-stage user request scheme plays an efficient role in average retrieval delay reduction. Solving optimization problems for optimal two-stage request scheme determines the proper number of packets requested in the first stage and follows high decoding probability.

3 Methods

3.1 LT codes

LT codes [18] are the first class of universal fountain codes. These codes can potentially generate infinite encoded symbols through the XOR operation of a subset of original symbols. For every encoded symbol, a degree d is chosen independently from a given degree distribution. LT codes can recover k original symbols from any $k(1 + \epsilon)$ encoded symbols with probability $1 - \delta$. where ϵ is known as overhead, the number of encoded symbols is equivalent to $k + O(\sqrt{k} \cdot \ln^2(k/\delta))$ and δ indicates the allowable failure probability of decoding. Belief propagation (BP) is used as an efficient decoding algorithm for these near-optimal codes which depends on degree-1 encoding symbols [19]. In LT code-based cloud storage system, a user file is first fragmented into k original symbols, and then, these original symbols are encoded into n encoded symbols. We briefly describe the encoding and BP decoding procedure of LT codes for $k = 4$ and $n = 5$ in Figs. 1 and 2, respectively.

A number of original symbols that are combined together is known as code degree, which is designated from two common degree distributions. First, ideal soliton distribution (ISD) $\rho(i)$ is defined as follows

$$\rho(i) = \begin{cases} \rho(1) = 1/k \\ \rho(i) = 1/i(i-1) & i = 2, \dots, k \end{cases} \quad (1)$$

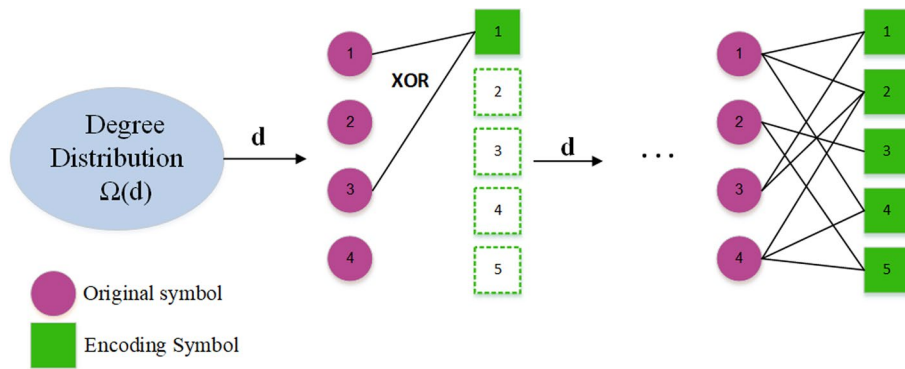


Fig. 1 LT encoding process

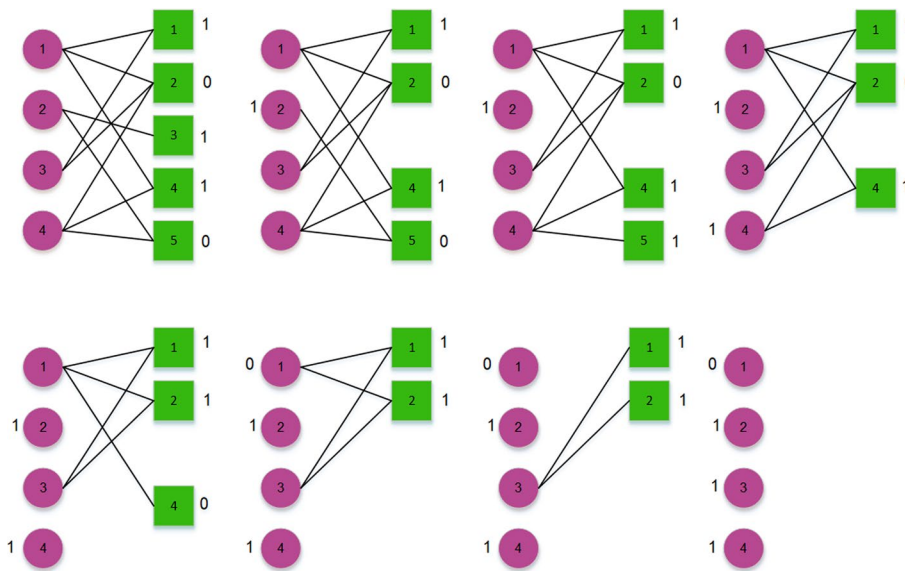


Fig. 2 Belief propagation decoding

One of the main goals in the design of good degree distribution is the ripple size. Ripple is a set of covered original symbols that have not been processed yet. If all the original symbols are covered, retrieval is successful while the process fails when the ripple is not empty at the end of the retrieval. The ripple size should be kept as small as possible to prevent redundant coverage of original symbols. On the other hand, the ripple should be large enough to avoid the disappearance of the ripple until the end of the process. This ideal size is called the expected ripple size that is too small with ISD, also fragile confronting any variation. Although ISD performs weekly in practice due to its ripple expected size that is one, it provides great perception for new distributions. The main distribution of LT codes is robust soliton that is denoted by $\mu(i)$. Let $R = c \cdot \ln(k/\delta) \sqrt{k}$ expected ripple size for some suitable constant $c > 0$. Define

$$\tau(i) = \begin{cases} R/ik & i = 1, \dots, k/R - 1 \\ R \ln(R/\delta)/k & i = k/R \\ 0 & i = k/R + 1, \dots, k \end{cases} \quad (2)$$

Add $\rho(i)$ to $\tau(i)$ and normalize to obtain $\mu(i)$

$$\beta = \sum_{i=1}^k \rho(i) + \tau(i) \quad (3)$$

$$\mu(i) = (\rho(i) + \tau(i))/\beta \quad i = 1, \dots, k \quad (4)$$

With a good degree distribution, LT codes can perform well. Although robust soliton distribution ensures ripple does not disappear during decoding process with high probability and can achieve good performance for LT codes, newly introduced distributions can improve performance in terms of overhead and recovery probability which is considerable, since cloud storage follows the “pay-as-you-use” paradigm.

3.2 Poisson robust soliton distribution

By combining the characteristics of Poisson distribution (PD) and robust soliton distribution (RSD), recently introduced distribution [20] with appropriate parameters can generate more degree-1 compared with RSD. Thus, PRSD successful retrieval probability outperforms RSD in lower overhead. The improved PD (IPD) is given by

$$\theta(i) = \begin{cases} 1/2 & i = 2 \\ \lambda^i e^{-\lambda}/i! & i = 1, 3, \dots, k \end{cases} \quad (5)$$

where λ is a positive constant.

Then, the proposed PRSD is obtained as follows

$$\Omega(i) = \theta(i) + \tau(i) / \sum_{i=1}^k \theta(i) + \tau(i) \quad i = 1, 2, \dots, k \quad (6)$$

PRSD provides lower average degrees and limited degrees in comparison with RSD. Thus, we can achieve cloud storage with faster retrieval and higher successful decoding probability in lower overheads.

3.3 Combined Poisson robust soliton distribution

In order to reduce overhead and consuming the time of encoding and decoding process, CPRSD proposed by combining IPD and RSD is as follows [21]

First, $\eta(i)$ is obtained from a normalization of $\theta(i)$,

$$\eta(i) = \theta(i) / \sum_{d=1}^k \theta(i) \quad i = 1, 2, \dots, k \quad (7)$$

The CPRSD is represented as

$$\Lambda(i) = \eta(i).a + \mu(i).(1 - a) \quad (8)$$

where the range of a is located between 0 and 1.

4 System model

To investigate the performance of LT codes with new distributions on a cloud storage system, we consider a small-scale cloud with 15 storage nodes. First, we encode 20 user data of various sizes with LT codes, and then, the distribution of encoded data is accomplished over cloud storage in regional and multi-regional storage modes [22]. In regional mode, data are distributed over a region with at least two availability zones, and region selection is based on minimum distance to data owner location. In the multi-regional mode that at least two regions are selected, we assume selection of the first region as the nearest one and random selection of the second region that can be equivalent to user options in cloud storage systems. To resemble the function of our system model to the reality of cloud storage system and also study retrieval time, we consider M/G/1 queue for every storage node [6] and a further M/G/1 queue for the head server to direct retrieval requests to corresponding nodes.

We inspect the successful decoding probability of LT code-based cloud storage with PRSD and CPRSD degree distributions in two state. The first one is a non-removal state of storage nodes, and the latter is a removal state to check the effect of inaccessibility or failure of nodes. The general structure of our system model is shown in Fig. 3.

5 Simulation results and discussion

5.1 Parameter selection

In this paper, we study LT codes with $k = \{100, 250, 500\}$. All simulations run in MATLAB. The first step is selecting n , the number of encoded symbols that are required to recover original symbols with high probability. As shown in Table 1, obtained values of n corresponding to k in our system model are almost close to values derived from the empirical model for decodability in [17] and the term $k + O(\sqrt{k} \cdot \ln^2(k/\delta))$ [18].

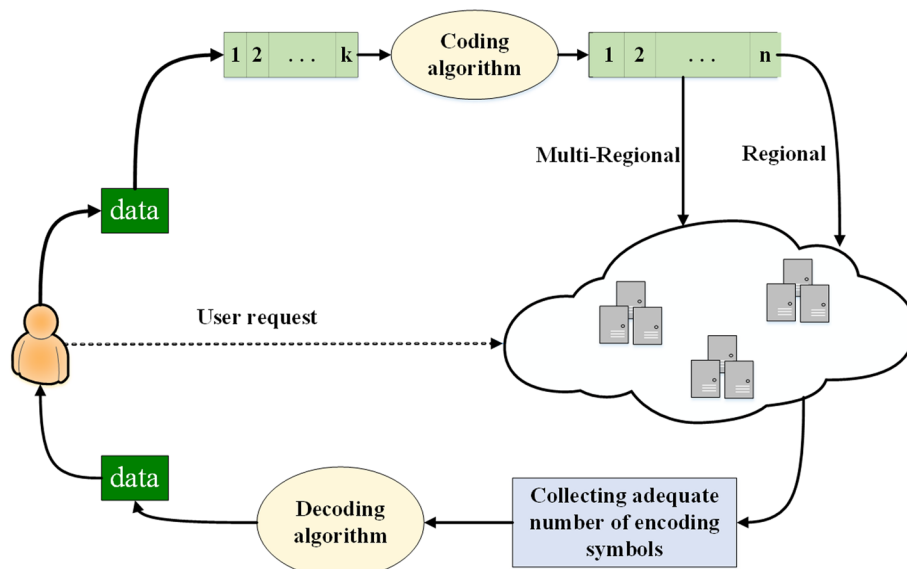


Fig. 3 General structure of our LT code-based cloud storage

Table 1 Required number of encoding symbols for successful decoding

k	[17]	$k + O(\sqrt{k} \cdot \ln^2(k/\delta))$	Simulation
100	165	157	158
200	310	271	290
250	–	333	388
300	450	395	441
400	580	518	568
500	–	639	710

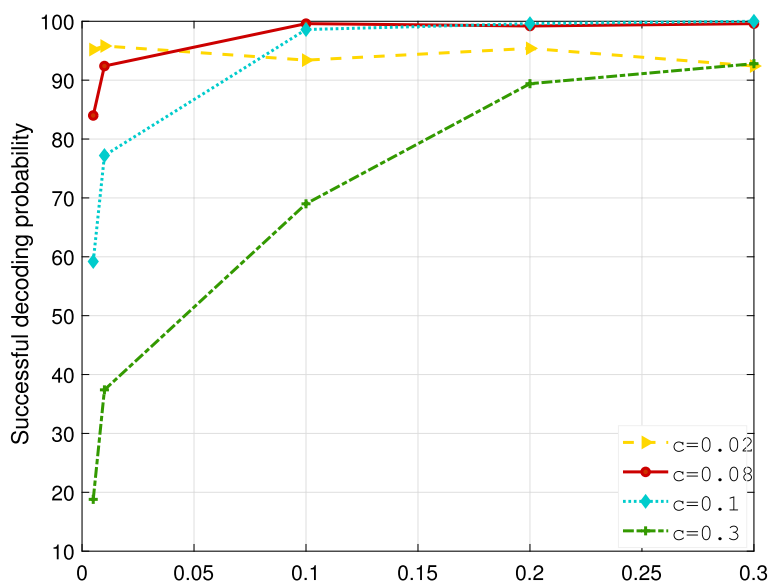


Fig. 4 Successful decoding probability with RSD for different c and δ

At the second step, two main parameters of RSD and PRSD, c and δ are determined that play important roles in degree distributions achievement. We investigate how the factors c and δ affect the probability of successful retrieval to discover a mutually suitable pair for both RSD and PRSD. As illustrated in Figs. 4 and 5 for $k = 250$, a higher successful decoding probability of RSD takes place within the range $[0.1, 0.3]$ of δ and two values 0.08 and 0.1 for parameter c . In addition, better performance of PRSD can be observed for $c = 0.08$ and δ measures located in the range of $[0.01, 0.1]$. Thus, we set $c = 0.08$ and $\delta = 0.1$ to reach the highest probable overall performance for both RSD and PRSD, which also stands for other values of k . The measure of failure decoding probability δ is reasonable in practice.

Finally, a as a fundamental parameter of CPRSD is selected to represent the contribution of IPD and RSD, in new degree distribution. Successful decoding probability against a for different k and versus overhead for different a is displayed in Figs. 6 and 7, respectively. As shown in Fig. 6, we can achieve higher successful decoding probabilities by the range $[0.3, 0.5]$ of parameter a for various values of original symbols. Since the variation of a has an effect on overhead as well as successful decoding probability, a trade-off is discussed. The trade-off offers the highest possible

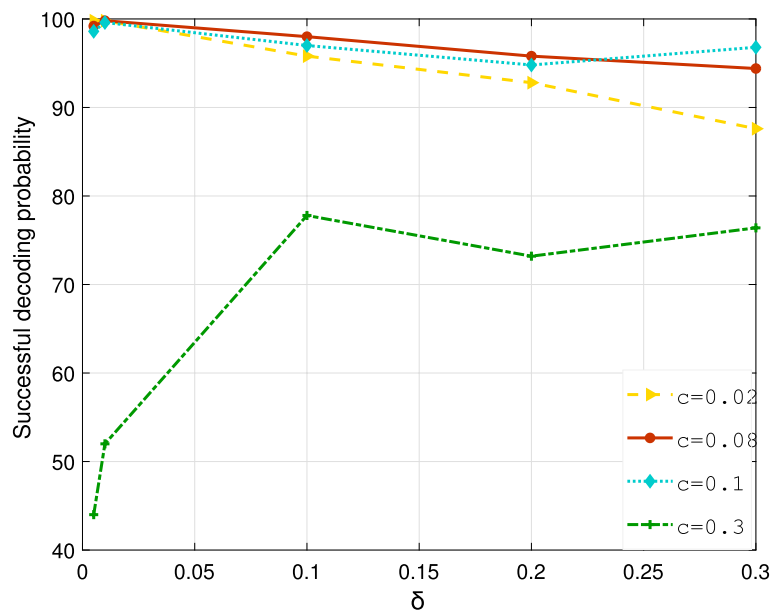


Fig. 5 Successful decoding probability with PRSD for different c and δ

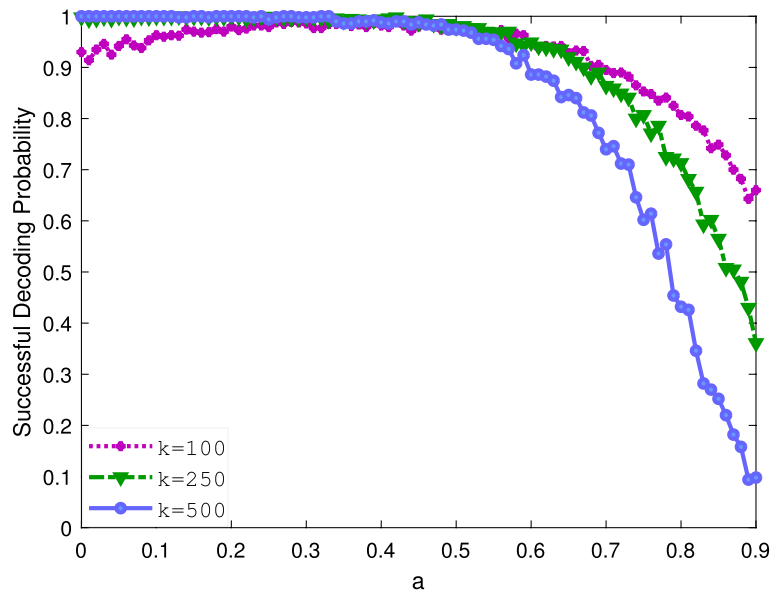


Fig. 6 Successful decoding probability with CPRSD for different k and a

probability of decoding, while the overhead is kept as small as possible. We set $a = 0.4$ to reach an acceptable compromise between overhead and successful decoding probability, which means the additional contribution of RSD provides much improvement. Based on the analysis of expectations and similarity of mathematical properties of PD and BD when $k < 20$, we consider $\lambda \approx 3.04$ [20].

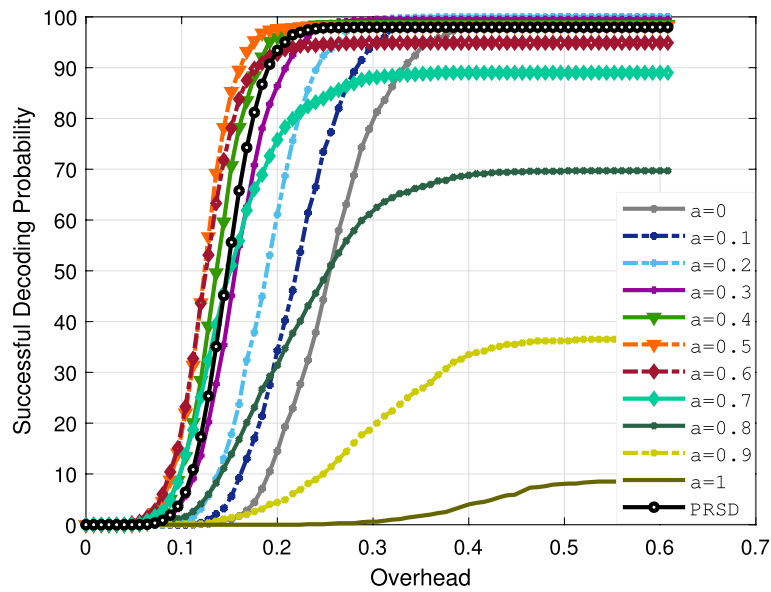


Fig. 7 Successful decoding probability with CPRSD for $k = 250$ and different a

5.2 Theoretical analysis

As mentioned before, exploiting PRSD and CPRSD leads to faster retrieval and higher successful decoding probability in lower overheads in comparison with RSD. We provide a few theoretical analyses of some indicators to prove our claims.

As RSD is the combination of $\rho(i)$ and $\tau(i)$, PRSD is constructed by $\theta(i)$ and $\tau(i)$, also the combination of all these distributions generates CPRSD, average degree, degree-one, maximum degree, and the number of encoding symbols is studied approximately through the expectations of $\rho(i)$ and $\theta(i)$ as follows.

First, one of the parameters which have an essential role in the retrieval process is the *average degree*. The average degree of encoding symbols should be as few as possible.

For RSD:

$$E(\rho(i)) = \sum_{i=1}^k i\rho(i) = 1/k + \sum_{i=2}^k 1/i(i-1) \leq 1/k + H(k) \tag{9}$$

For PRSD:

$$E(\theta(i)) = \sum_{i=1}^k i\theta(i) = \lambda e^{-\lambda} + 1 + \lambda e^{-\lambda} \sum_{i=3}^k \lambda^{i-1}/(i-1)! < 1 + \lambda + \lambda e^{-\lambda} \tag{10}$$

Finally, the average degree of CPRSD is $E(a\theta(i) + (1-a)\rho(i))$, and therefore,

$$aE(\theta(i)) + (1-a)E(\rho(i)) < a(1 + \lambda + \lambda e^{-\lambda}) + (1-a)(1/k + H(k)) \tag{11}$$

where $0 < a < 1$.

Based on the analysis mentioned in the article, we consider $\lambda \approx 3.04$, so

$$E(\theta(i)) < 4.1854 \tag{12}$$

By increasing the number of original symbols k , the average degree of $\rho(i)$ tends to $H(k)$. As $\ln(k) < H(k) < \ln(k) + 1$, we have

$$\forall k \geq 25 \quad E(\theta(i)) < E(\rho(i)) \tag{13}$$

In addition, for $k \geq 25$

$$aE(\theta(i)) + (1 - a)E(\rho(i)) < E(\rho(i)) \tag{14}$$

According to the terms stated above, for $k = \{100, 250, 500\}$ the average degree of PRSD and CPRSD is smaller than RSD. Less average degree is equivalent to fewer XOR operations required in the encoding and decoding process that can be led to faster retrieval. Second parameter that shows the superiority of PRSD and CPRSD over RSD is the *number of degree-one encoded symbols*. PRSD and CPRSD provide a higher fraction of degree-one encoding symbols due to the nature of Poisson distribution with appropriate parameter selection. Since the ripple is a set of degree-one encoding symbols in the decoding process, a large expected ripple size at the beginning of the process brings higher successful decoding probability when the decoder receives fewer encoding symbols. The number of degree-one encoding symbols for three distributions are given, respectively,

$$\begin{aligned} \mu(1) &\longrightarrow \rho(1) = 1/k \\ \Omega(1) &\longrightarrow \theta(1) = 1/2 \\ \Lambda(1) &\longrightarrow a\theta(1) + (1 - a)\rho(1) \end{aligned}$$

For $k > 2$

$$\theta(1) > \rho(1) \tag{15}$$

Also, $\forall k > 2$, we have

$$a\theta(1) + (1 - a)\rho(1) > \rho(1) \tag{16}$$

In addition, we present a *maximum degree* that indicates PRSD and CPRSD which outperform RSD regarding better retrieval in lower overheads and time-consuming decoding process. Degree distribution tends to zero when the degree approaches the maximum degree that can be generated. To study a maximum degree, we define ϵ' instead of zero which is considered small enough. For RSD,

$$1/i(i - 1) \geq \epsilon' \tag{17}$$

As the degree is a positive integer,

$$i \geq 1 + \sqrt{1 + 4/\epsilon'}/2 \tag{18}$$

By considering $\epsilon' \leq 0.001$, the maximum degree of RSD is obtained as

$$i_{max} \geq 33 \tag{19}$$

For PRSD,

$$e^{-\lambda} \lambda^i / i! \geq \epsilon' \tag{20}$$

Through Stirling's approximation of $\ln i! \approx i \ln i - i$,

$$i(\ln\lambda - \ln i + 1) \geq \lambda + \ln\epsilon' \tag{21}$$

To reach the maximum degree, we exploit determining the sign of the above term. Since the degree i is a positive integer and the right expression is negative based on $\epsilon' \leq 0.001$ and $\lambda \approx 3.04$, the left expression should be located between -1 and 0 to guarantee maximum degree and negativity.

$$i \leq \frac{\lambda + \ln\epsilon'}{\ln\lambda - \ln i + 1} \tag{22}$$

$$-1 < \ln\lambda - \ln i + 1 < 0 \tag{23}$$

Thus,

$$9 \leq i \leq 22 \tag{24}$$

and $i_{max} = 22$.

In the following for CPRSD, we exert the range of parameters a on the term below

$$a(e^{-\lambda}\lambda^i/i!) - (1 - a)(1/i(i - 1)) \geq \epsilon' \tag{25}$$

$$0 < \frac{\epsilon' - 1/i(i - 1)}{e^{-\lambda}\lambda^i/i! - 1/i(i - 1)} < 1 \tag{26}$$

As $i! \geq i(i - 1)$, we determine the sign of denominator, and thus,

$$\forall i > 2.7342 \quad e^{-\lambda}\lambda^i/i! - 1/i(i - 1) < 0 \tag{27}$$

Also $\epsilon' - 1/i(i - 1)$ should be negative, and therefore,

$$3 \leq i \leq 32 \tag{28}$$

and $i_{max} = 32$.

Thus, two new defined distributions provide a smaller maximum degree in comparison with RSD.

The last parameter to investigate is the *number of encoding symbols*. For LT codes with RSD, the number of encoding symbols is $k + O(\sqrt{k} \cdot \ln^2(k/\delta))$ [18] that is obtained from

$$n_{RSD} \leq k + R.H(k/R) + R.\ln(R/\delta) \tag{29}$$

By neglecting the terms corresponding to mutual distribution $\tau(i)$, we have the number of encoding symbols for PRSD and CPRSD as follows

$$n_{PRSD} = k \cdot \left(\sum_{i=1}^k \theta(i) \right) = k \cdot (\lambda e^{-\lambda} + 1/2 + \sum_{i=3}^k \lambda^i e^{-\lambda}/i!) \tag{30}$$

According to $\sum_{i=0}^{\infty} \lambda^i/i! = e^\lambda$,

$$k \cdot (\lambda e^{-\lambda} + 1/2 + \sum_{i=3}^k \lambda^i e^{-\lambda}/i!) < k \tag{31}$$

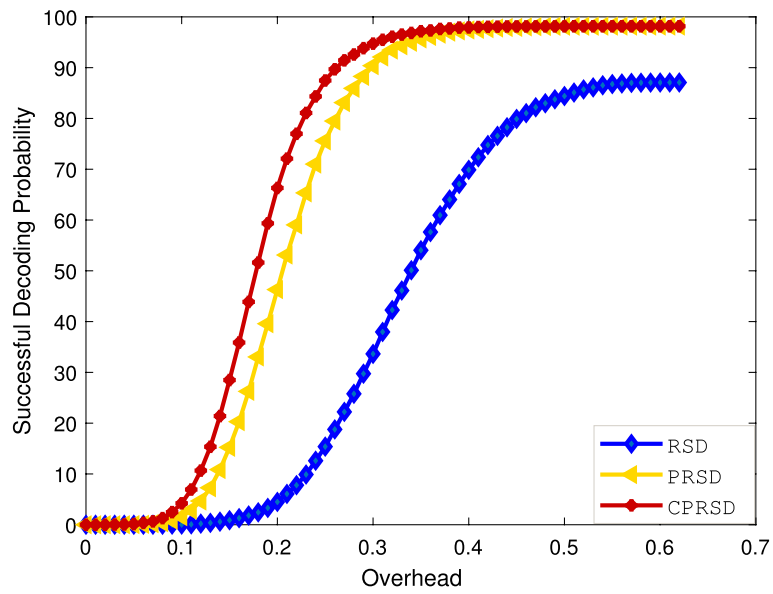


Fig. 8 Successful decoding probability for $k = 100$ in non-removal state

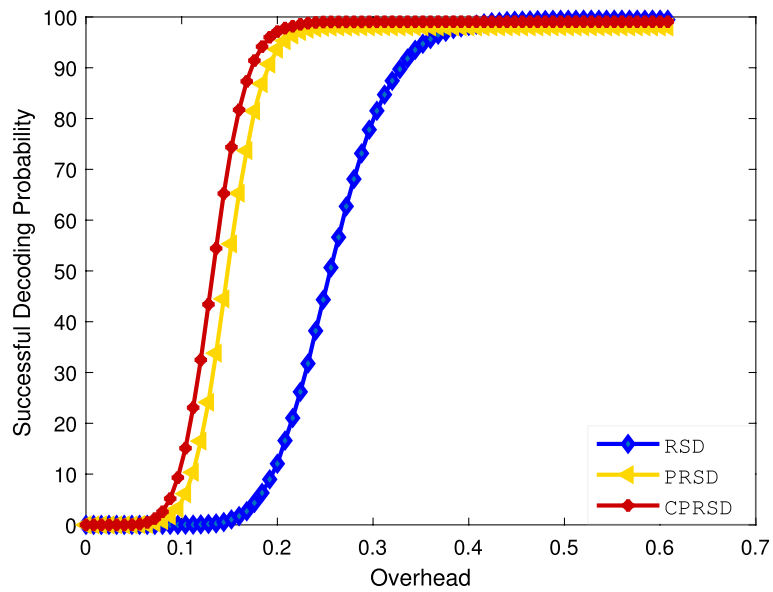


Fig. 9 Successful decoding probability for $k = 250$ in non-removal state

As the maximum measure of $\lambda^2 e^{-\lambda} / 2$ that happens at $\lambda = 2$ is near the probability of $\theta(2)$, we have

$$n_{PRSD} < n_{RSD} \tag{32}$$

Consider the number of encoding symbols for CPRSD as

$$n_{CPRSD} = k \cdot (a \cdot \sum_{i=1}^k \theta(i) + (1 - a) \cdot \sum_{i=1}^k \rho(i)) \tag{33}$$

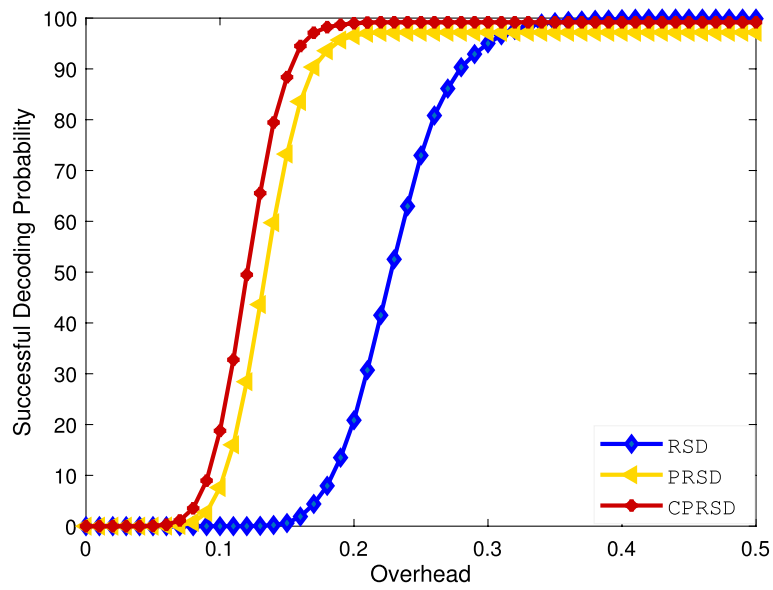


Fig. 10 Successful decoding probability for $k = 500$ in non-removal state

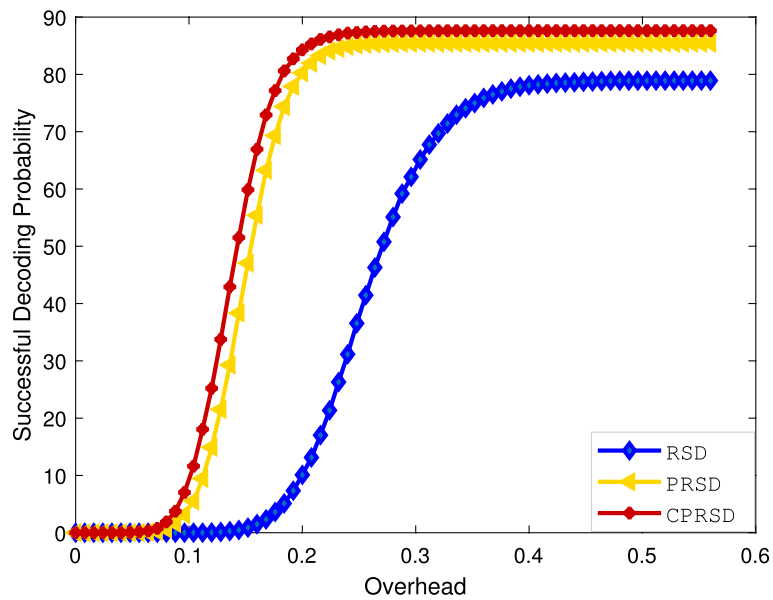


Fig. 11 Successful decoding probability for $k = 250$ in removal state

Through the inequality acquired for n_{PRSD} , we can easily reach the following expression

$$n_{CPRSD} < n_{RSD} \tag{34}$$

In conclusion, PRSD and CPRSD provide faster and more successful retrieval, while the decoder receives fewer encoding symbols. The numerical results in the next sections confirm our claims.

5.3 Degree distributions comparison

In this section, we present the comparison of successful decoding probability against overhead, for RSD, PRSD, and CPRSD on a cloud storage system. Overhead is defined as follows

$$\epsilon = (n - k)/k \quad (35)$$

To study LT code-based cloud storage performance, we consider two phases through simulations due to the random nature of encoding and decoding of LT codes and also the validation of our results. One hundred repeats are set for the outer phase, while each outer phase experiences one hundred repeats of the inner phase. The outer phase includes a selection of encoded symbols degree, generating metadata, encoding by LT codes, and finally distributing over the cloud. Furthermore, the inner phase is assumed in every outer phase which encompasses users' requests for various data from diverse geographical locations at random and data retrieval process. As shown in Figs. 8, 9 and 10, for $k = \{100, 250, 500\}$ and in the non-removal state of storage nodes increasing the number of original symbols, the performance of considered distributions becomes close together. RSD needs more overhead to retrieve successfully, which means much retrieval time and more storage space. Thus, successful decoding probability with CPRSD and PRSD outperforms RSD in particular at lower overheads.

We study the behavior of proposed cloud storage with three distributions in confronting unavailability or loss of encoding symbols. The goal accomplished considering one data center is out of reach randomly in every simulating iteration. Figure 11 depicts the successful decoding probability for $k = 250$ in the removal state. Therefore, better robustness and higher successful decoding probability can be achieved by applying CPRSD and PRSD for the various number of original symbols.

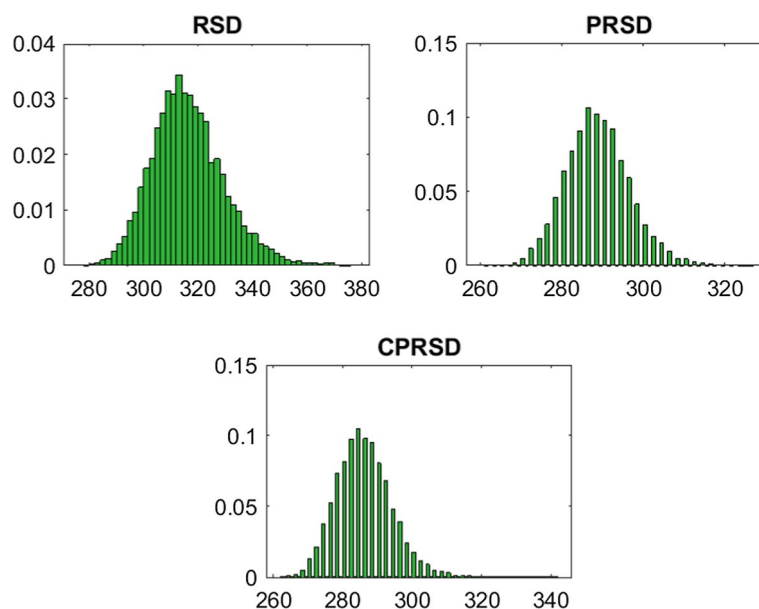


Fig. 12 Histogram of required number of encoding symbols for successful data retrieval for $k = 250$

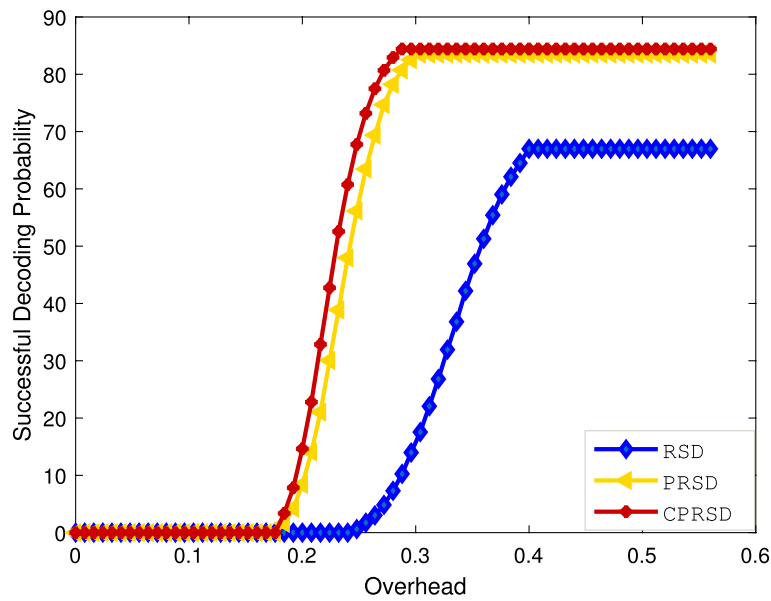


Fig. 13 Successful decoding probability with proposed decoding process for $k = 250$ in removal state

Table 2 Retrieval time comparison between main and proposed decoding process

k	Retrieval time (second)					
	RSD		PRSD		CPRSD	
	Main	Proposed	Main	Proposed	Main	Proposed
100	0.4510	0.2530	0.2962	0.1606	0.4403	0.2488
250	4.7818	1.5613	1.9538	0.6583	3.9001	1.3413
500	9.7998	4.8760	5.0312	1.4730	14.5405	4.8298

Successful data retrieval in lower overheads and also in the presence of partly encoding symbols loss is notable attainment in cloud storage systems. Furthermore, it could lead to less storage space, retrieval time, and cost of users and providers, hence more satisfactory services.

5.4 Time improvement

We assume a scenario to compute data retrieval time. First retrieval time is defined as follows

$$T_{total} = T_w + \frac{L}{r} + T_{decoding} \tag{36}$$

T_w is queuing delay, the second term is data transmission time, and $T_{decoding}$ is the mean of decoding process time in simulations.

As mentioned before, we consider M/G/1 queue for every storage node and head datacenter. Our simulations run based on the following assumptions. In every iteration, selection accomplishes randomly among 2, 4, 6^{ms} for mean service time of data

centers and among $1, 1.5, 2^{ms}$ for variance. The mean arrival rate is assumed 5^{ms} . Moreover, the arrival rate for the head data center is set to 10, and mean and variance of service time are considered as 1 and 3^{ms} , respectively. The factor r is assumed 30.03 Mbps, and rests on the global average download speed report for mobile Internet in 2019 [23].

Generally, the LT decoder needs an undetermined number of encoded symbols from the storage system to retrieve data successfully. More delay arises from this ambiguity in LT code-based cloud storage systems. Thus, there is a compromise between successful decoding probability and retrieval delay. According to our observation during the decoding process, the number of encoding symbols required for successful data retrieval follows normal distribution as shown in a histogram for $k = 250$ in Fig. 12. Since retrieval time is comparable to the user experience of cloud storage service, we design a scheme in which the decoding process is implemented for the number of encoding symbols lying within two standard deviations from the mean instead of blind search in the almost big interval. Figure 13 shows successful decoding probability against overhead after applying the proposed decoding process in the removal state.

As clearly seen, reduction in successful decoding probability is negligible in particular for PRSD and CPRSD, whereas time reduction is significant according to Table 2. Retrieval time using the proposed decoding process can be decreased up to 70 percent with PRSD and 67 percent with CPRSD.

6 Conclusion

In this paper, we studied LT code-based cloud storage using newly designed degree distributions. Data retrieval achieves much success with PRSD and CPRSD compared with the conventional solution, RSD, specifically in smaller overheads, moreover in the presence of unavailability or loss of encoding symbols. Furthermore, we proposed a modified decoding algorithm in order to obtain retrieval time improvement. The performance analysis and experimental results show that the proposed LT code-based cloud storage system can provide higher successful decoding probability, less storage space, more robustness, and faster data retrieval.

Abbreviations

RSD	Robust soliton distribution;
PRSD	Poisson robust soliton distribution;
CPRSD	Combined Poisson robust soliton distribution;
IDC	International Data Corporation;
LRC	Local reconstruction codes;
MDS	Maximum separable codes;
BLCS	Block-level cloud storage;
BP	Belief propagation;
PD	Poisson distribution;
IPD	Improved PD;

Acknowledgements

Not applicable.

Author contributions

Dr. Seyed Masoud Mirrezaei, Prof. Ghosheh Abed Hodtani, and Nastaran Chakani contributed to conceptualization; Dr. Seyed Masoud Mirrezaei and Nastaran Chakani were involved in formal analysis and funding acquisition; Nastaran Chakani contributed to investigation and writing—original draft, and provided resources and software; Dr. Seyed Masoud Mirrezaei was involved in methodology; and Dr. Seyed Masoud Mirrezaei and Prof. Ghosheh Abed Hodtani contributed to project administration, supervision, and writing—review editing. All authors read and approved the final manuscript.

Author informations

Nastaran Chakani received the BSc degree in communication engineering from Azad University of Mashhad, Mashhad, Iran, in 2012 and the MSc degree in communications engineering from the Shahrood University of Technology, Shahrood, Iran, in 2018. Her current research interests include information theory, channel and source coding, cloud storage, and information-theoretic learning.

Seyed Masoud Mirrezaei received his BSc degree in Communication Engineering from K. N. Toosi University of Technology in September 2004. He received his MSc and PhD degrees in Electrical Engineering from Amirkabir University of Technology, Tehran, Iran, in 2007 and 2013, respectively. He has been as a PhD student and researcher in "Mobile and Wireless Networks Research Laboratory" in Amirkabir University of Technology under the supervision of Prof. Karim Faez. Also, he was a visiting research student in the Signal Design and Analysis Laboratory (SDAL) at Queen's University, Kingston, Canada, under the supervision of Prof. Shahram Yousefi from March 2011 to February 2012. He joined the Electrical Engineering Department of Shahrood University of Technology from 2013 up to now. His research interests lie in communications, cloud systems, big data, networks, information theory, signal processing, channel coding, and network coding.

Ghosheh Abed Hodtani received the BSc degree in electronics engineering and the MSc degree in communications engineering, both from Isfahan University of Technology, Isfahan, Iran, in 1985 and 1987, respectively. He joined Electrical Engineering Dept., at Ferdowsi University of Mashhad, Mashhad, Iran, in 1987. He decided to pursue his studies in 2005 and received the PhD degree (with excellent grade) from Sharif University of Technology, Tehran, Iran, in 2008, and he is now a professor in electrical engineering. His research interests are in multi-user information theory, communication theory, wireless communications, information-theoretic learning, and signal processing. Prof. Hodtani is the author of a textbook on electrical circuits and is the winner of the best paper award at IEEE ICT-2010

Funding

Not applicable.

Availability of data and materials

Not applicable

Declarations

Consent for publication

All authors have agreed and given their consent for submission of this paper to EURASIP Journal of Wireless Communications and Networking.

Competing interests

The authors declare that they have no competing interests.

Received: 7 June 2021 Accepted: 4 June 2022

Published online: 21 June 2022

References

1. W. paper IDC, The digitization of the world from edge to core (2018)
2. H. Weatherspoon, J.D. Kubiatowicz, Erasure coding vs. replication: a quantitative comparison. *Lecture Notes Comput. Sci.* **2429**, 328–337 (2002). https://doi.org/10.1007/3-540-45748-8_31
3. P. Bhuvaneshwari, C. Tharini, Review on LDPC codes for big data storage. *Wirel. Pers. Commun.* **117**(2), 1601–1625 (2021)
4. G. Joshi, Y. Liu, E. Soljanin, Coding for fast content download. In: 2012 50th Annual allerton conference on communication, control, and computing (Allerton), pp. 326–333. IEEE (2012)
5. C. Huang, H. Simitci, Y. Xu, A. Ogun, B. Calder, P. Gopalan, J. Li, S. Yekhanin, Erasure coding in windows Azure storage. In: Proceedings of the 2012 USENIX annual technical conference, pp. 15–26 (2012)
6. G. Joshi, Y. Liu, E. Soljanin, On the delay-storage trade-off in content download from coded distributed storage systems. *IEEE J. Select. Areas Commun.* **32**(5), 989–997 (2014)
7. A. Kumar, R. Tandon, T.C. Clancy, On the latency and energy efficiency of distributed storage systems. *IEEE Trans. Cloud Comput.* **5**(2), 221–233 (2017). <https://doi.org/10.1109/TCC.2015.2459711>
8. A.G. Dimakis, P.B. Godfrey, Y. Wu, M.J. Wainwright, K. Ramchandran, Network coding for distributed storage systems. *IEEE Trans. Inf. Theory* **56**(9), 4539–4551 (2010)
9. A.G. Dimakis, K. Ramchandran, Y. Wu, C. Suh, A survey on network codes for distributed storage. *Proc. IEEE* **99**(3), 476–489 (2011)
10. J. Huang, Z. Fei, C. Cao, M. Xiao, Design and analysis of online fountain codes for intermediate performance. *IEEE Trans. Commun.* **68**(9), 5313–5325 (2020)
11. H. Xia, A.A. Chien, RobuStore: a distributed storage architecture with robust and high performance. In: Proceedings of the 2007 ACM/IEEE conference on supercomputing, pp. 10–16. <https://doi.org/10.1145/1362622.1362682>
12. N. Cao, S. Yu, Z. Yang, W. Lou, Y.T. Hou, LT codes-based secure and reliable cloud storage service. In: 2012 Proceedings IEEE INFOCOM, pp. 693–701. IEEE (2012)
13. J. Huang, Z. Fei, C. Cao, M. Xiao, X. Xie, Reliable broadcast based on online fountain codes. *IEEE Commun. Lett.* (2020)
14. M. He, C. Hua, W. Xu, P. Gu, X.S. Shen, Delay optimal concurrent transmissions with raptor codes in dual connectivity networks. In: IEEE Transactions on network science and engineering (2021)
15. P. Shi, Z. Wang, D. Li, W. Xiang, Zigzag decodable online fountain codes with high intermediate symbol recovery rates. *IEEE Trans. Commun.* **68**(11), 6629–6641 (2020)

16. C. Anglano, R. Gaeta, M. Grangetto, Exploiting rateless codes in cloud storage systems. *IEEE Trans. Parallel Distrib. Syst.* **26**(5), 1313–1322 (2015). <https://doi.org/10.1109/TPDS.2014.2321745>
17. H. Lu, C.H. Foh, Y. Wen, J. Cai, Delay-optimized file retrieval under LT-based cloud storage. *IEEE Trans. Cloud Comput.* **5**(4), 656–666 (2015)
18. M. Luby, LT codes. In: *Proceedings of the 43rd annual IEEE symposium on foundations of computer science, 2002*. pp. 271–280. IEEE (2003)
19. J. Thorpe, Low-density parity-check (LDPC) codes constructed from protographs. *IPN Progr. Rep.* **42**(154), 42–154 (2003)
20. W. Yao, B. Yi, T. Huang, W. Li, Poisson robust soliton distribution for LT codes. *IEEE Commun. Lett.* **20**(8), 1499–1502 (2016). <https://doi.org/10.1109/LCOMM.2016.2578920>
21. W. Yao, B. Yi, W. Li, T. Huang, Q. Xie, CPRSD for LT codes. *IET Commun.* **10**(12), 1411–1415 (2016). <https://doi.org/10.1049/iet-com.2015.1183>
22. <https://cloud.google.com/storage/docs/storage-classes>. Storage-classes: No Title
23. J. Clement: average mobile fixed broadband download upload speeds. <https://www.statista.com/statistics/896779/average-mobile-fixed-broadband-download-upload-speeds>

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Submit your manuscript to a SpringerOpen[®] journal and benefit from:

- ▶ Convenient online submission
- ▶ Rigorous peer review
- ▶ Open access: articles freely available online
- ▶ High visibility within the field
- ▶ Retaining the copyright to your article

Submit your next manuscript at ▶ [springeropen.com](https://www.springeropen.com)
