Journal of Electrical Engineering, Vol. 75, No. 1, 2024, pp. 72-76

sciendo

PAPERS

Linear block code with locality and availability inspired by tetrahedron

Peter Farkaš[1,2*]

The primary application of codes with locality and availability in distributed storage is for data recovery in case that data are lost on some damaged servers. Locality enables the recovery of lost data by contacting only a restricted number of remaining servers. Availability means that more than one subset of servers providing locality is available for data recovery for each server. The secondary application of these codes in distributed storage is to allow access to hot data in times of high demand. In this paper it is shown that the binary linear [14, 4, 7] code has locality 2 and availability 6 and it can be interpreted as a three-dimensional graph obtained from a [7, 3, 4] Simplex code. It is achieving upper bounds on basic parameters for codes with all-symbols locality and availability. This code can be a building element of more complex codes with scalability inspired by three-dimensional structures. The availability spectrum is introduced as a tool for analyzing codes with locality and availability.

Keywords: data integrity, data recovery, distributed data storage, error correction, linear code

1 Introduction

Currently and also in the foreseeable future, many information infrastructures and applications will rely on distributed storage systems. For example, cloud applications, autonomous vehicles and Industry 4.0 will need low delay access to data in such systems.

On the other hand, it is well known that the data in distributed storage systems or in storage systems generally are sometimes lost and need to be recovered. Therefore, redundant servers (nodes) are necessary which can help to recover the data from lost servers in case of damage. The number of redundant nodes is one of the main cost burdens in data centers. It depends on the technology used and therefore, simple data replication which needs high redundancy was long ago substituted by more efficient approaches which are based on redundant codes.

For example, in RAID technology, Reed Solomon (RS) codes are used for data recovery. RS codes are so called maximum distance separable (MDS) codes. MDS means that for RS codes with codeword length n and dimension (number of information symbols) k the code distance d_m is equal to $n - k + 1$. Such a code distance allows correcting up to $n - k$ erased symbols in each codeword of the RS code. Therefore, RS codes allow minimizing the number of redundant nodes in distributed storage systems.

Note 1: An erased symbol is one whose value (contents) is lost but whose position inside the codeword is known.

Note 2: Linear block codes are often denoted as [n, k, d_m] codes.

However, especially in distributed storage systems it soon becomes clear that during recovery the cost of communication between distributed storage nodes is also another important cost burden. Consequently, it has to be minimized or at least decreased if possible.

Therefore, codes with locality have recently become popular because they allow recovering the lost data by procedures which need to download data only from a restricted number of local nodes. Further improvement is possible if the codes used behind the locality also have another property denoted as availability. Availability means that there are more disjoint subsets of nodes providing the locality. In practice the availability is important not only for data recovery but also for decreasing the access times to the so-called hot data if the demand is high.

In practical applications a compromise often has to be found between different demands in choosing a code such as reliability, cost, redundancy, locality, availability, response time, security, energy efficiency, CO_2 foot print, scalability and many others [1-5].

[1] Faculty of Electrical Engineering and Information Technology, Slovak University of Technology, Ilkovičova 3, 812 19 Bratislava, Slovakia
[2] Faculty of Informatics, Pan-European University, Tematínska 10, 851 05 Bratislava, Slovakia
p.farkas@ieee.org

https://doi.org/10.2478/jee-2024-0010, Print (till 2015) ISSN 1335-3632, On-line ISSN 1339-309X

Therefore, it is desirable that coding theory provides a broad palette of codes with locality, availability and other properties. It will allow practitioners to find better adjusted trade-offs between the many diverse requirements.

In [5] the codes with locality and availability were proposed which have also scalability. They were constructed based on graphs in a plane obtained from a [7, 3, 4] Simplex code. This invoked a question as to whether this Simplex code can help obtain new scalable codes with locality and availability based on 3-dimensional graphs.

In this paper it is shown that the [14, 4, 7] linear binary block code has not only maximal possible value of $d_m$for the other two parameters [8], but also has all-symbol locality and availability properties achieving upper bounds for its basic parameters in [9]. It can be interpreted as a result of merging the graph of four [7, 3, 4] Simplex codes drawn on tetrahedron. Therefore, in future it can be used as a building element of scalable codes with locality and availability based on 3-dimensional graphs.

The construction presented in this manuscript was inspired by [10], where in contrast to our approach each edge and vertex in the polyhedron corresponds to a codeword and parity check symbol of the code respectively.

The paper is organized as follows. In Section 2, a rudimentary introduction to linear block codes is made. In Section 3, the structure of the [14, 4, 7] code and its properties are analyzed and the results of this analysis are presented. In this section the availability spectrum is also introduced as a tool for analyzing codes with locality and availability. In conclusions the contributions of the paper are summarized.

2 Basic theory

Linear block code C is a k-dimensional subspace of an n-dimensional vector space over a finite field $GF(q)$.

The codeword $\boldsymbol{c}$ can be obtained as follows

$$\boldsymbol{c} = \boldsymbol{i}.\boldsymbol{G}\,, \tag{1}$$

where $\boldsymbol{G}$ is the generator matrix and $\boldsymbol{i}$ is an information vector. Its k coordinates are from $GF(q)$ and they contain the information. The rows in $\boldsymbol{G}$ are linear independent vectors over $GF(q)$.

If the finite field has only two elements, it is denoted as $GF(2)$ and the code constructed over it is a binary code. In the following text we will restrict our attention to binary codes only. We can therefore write a codeword as a vector

$$\boldsymbol{c} = (c_{n-1}, \quad \dots, \quad c_1, \quad c_0), \tag{2}$$

where $c_i \in GF(2)$; $\qquad i = 0, \quad 1, \quad \dots, \quad n-1.$

Each linear block code behind the $\boldsymbol{G}$ matrix can be also given by its parity check matrix $\boldsymbol{H}$. The rows of $\boldsymbol{H}$ are determined by control equations which have to fulfil each codeword or more exactly each subset of symbols in each codeword.

If the $\boldsymbol{H}$ matrix has $n-k>1$ rows, it is possible to form even more than $n-k$ control equations. Some of them are then linearly dependent. Each control equation allows computing a single symbol in it if the values of others are known.

The memory could be organized in such a way that a multiplicity of codes is used and in each server (node) only codeword symbols with the same index are stored as is illustrated in Fig. 1.

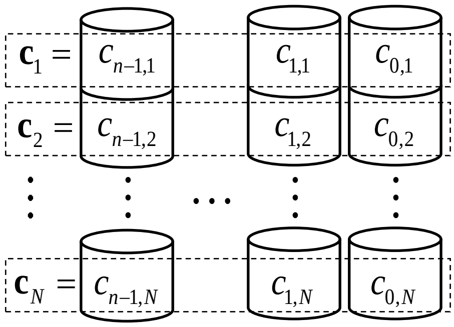

Fig. 1. Illustration how symbols of N codewords which protect information could be stored in n servers

This allows us to explain the connection between the theory which concentrates on analyzing the structure of the code taking into account particular symbols only and the nodes of distributed storages used in practice. If a server is damaged or lost in another way, we know its position but do not know the data which it stored. The data recovery using a code is then equivalent to erasure correction.

Codes with locality

Codes with locality allow correction of erased symbols using only a small number of other (local) symbols. More precisely, the i-th symbol c_i; $i = 0, 1, \dots, n-1$ of a codeword $\boldsymbol{c} \in C$, where C is an $[n, k, d_m]$ linear block code, has locality r if c_i can be recovered by accessing at most r other symbols from $\boldsymbol{c}$.

An $[n, k, d_m]$ code C has locality r and is denoted as r-LRC (locally recoverable code) if and only if all codeword symbols of all its codewords have locality r.

Codes with strict availability

Strict availability is a property which allows for recovering one erased codeword symbol in more than one way by accessing different disjoint sets of codeword symbols. More precisely, the i-th symbol c_i; $i = 0, 1, \ldots, n-1$ of a codeword $\boldsymbol{c} \in C$, where C is an $[n, k, d_m]$ linear block code, has strict availability t if and only if it can be recovered in t ways by accessing at least t disjoint sets of symbols (called repair sets) from a codeword $\boldsymbol{c}$.

The $[n, k, d_m]$ code is denoted (r, t)-LRC if all its symbols have locality r and strict availability t.

The binary Simplex Code with $n = 7$ and $k = 3$ is a nice and often used example of a (2, 3)-LRC [6].

Note 3: In this paper we will distinguish between strict availability and loose availability. In strict availability all repair sets are disjoint. In loose availability not all repair sets must be disjoint. Availability will denote strict or loose availability. The reason is that in some cases strict availability is not demanded for LRCs [7].

3 Analysis of the [14, 4, 7] code

The binary linear [14, 4, 7] linear binary block code is reaching the upper bound on code distance for its codeword length and dimension [8]. Therefore, it has minimal possible redundancy for these parameters and it can correct up to 3 errors or 6 erasures in each codeword.

This code could be obtained if we use the [7, 3, 4] Simplex code in such a way that we imagine that it is drawn on each face of a tetrahedron as illustrated in Fig. 2, where the vertices and edges of the graphs on the tetrahedron edges are merged.

For data repair it would be of interest to answer the questions about erasure correcting properties of this code expressed via locality and availability of its symbols. Therefore, in this section the locality and availability of the [14, 4, 7] code will be analyzed.

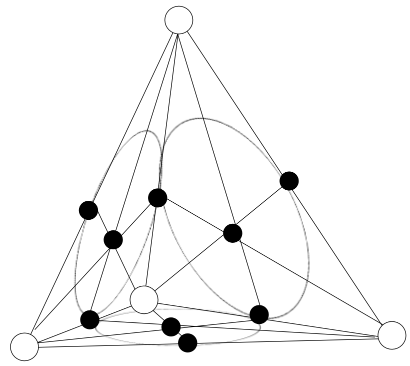

Fig. 2. The [14, 4, 7] code represented using a 3-dimensional graph drawn on tetrahedron faces with merged nodes on edges. White and black nodes represent information and parity bits respectively

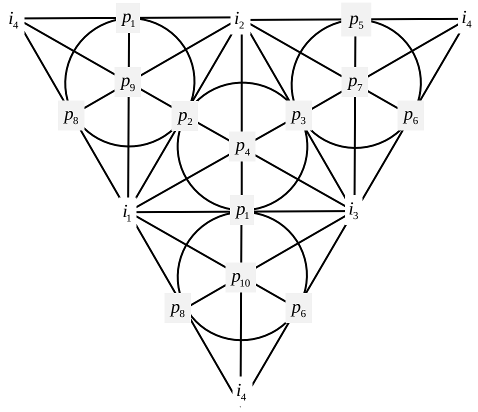

Fig. 3. Information and parity bits mapping on nodes in graph depicted in Fig. 2

Let us denote the codeword as

$$\boldsymbol{c} = (c_{13}, \quad \ldots, \quad c_1, \quad c_0) \tag{3}$$

and use the following mapping between the codeword symbols and information symbols and parity check symbols illustrated in Fig. 3:

$$c_{13} = i_1, \quad c_{12} = i_2, \quad c_{11} = i_3, \quad c_{10} = i_4, \tag{4}$$

$$c_9 = p_1, \quad c_8 = p_2, \quad c_7 = p_3, \quad c_6 = p_4, \quad c_5 = p_5, \tag{5}$$

$$c_4 = p_6, \quad c_3 = p_7, \quad c_2 = p_8, \quad c_1 = p_9, \quad c_0 = p_{10}, \tag{6}$$

Using this mapping we will obtain the following generator matrix $\boldsymbol{G}$ for the [14, 4, 7] code:

$$\boldsymbol{G} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \tag{7}$$

The parity check matrix $\boldsymbol{H}$ of this code is:

$$\boldsymbol{H} = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{8}$$

Observing Fig. 2 superficially (considering only control equations corresponding to the lines and circles drawn on the faces in Fig. 2) one can conclude that:

- Each information symbol has strict availability 6 with locality 2.
- Each parity check symbol drawn on each edge has strict availability 5 with locality 2.
- Each parity check symbol drawn in the center of gravity of the faces has strict availability 3 with locality 2.

However, a deeper analysis using computer revealed that:

- Each codeword symbol in [14, 4, 7] code has locality 2 and strict availability 6. Consequently, it is achieving upper bounds in [9] on k and d_m for codes with all-symbols locality 2, strict availability 6 and codeword length 14.

This computerized analysis was realized based on the following approach. All 1023 possible nonzero control equations were generated as linear combinations of rows in the parity check matrix $\boldsymbol{H}$ given by (8) and stored in matrix $\boldsymbol{A}$.

Note 4: The parity check matrix $\boldsymbol{H}$ of the original [14, 4, 7] code is at the same time the generator matrix of a [14, 10, 3] linear binary block code which is a dual code to the original [14, 4, 7] code. Therefore, all nonzero linear combinations of the rows from $\boldsymbol{H}$ are all nonzero codewords of the dual code.

Analyzing $\boldsymbol{A}$, it was discovered that each codeword symbol in the [14, 4, 7] code has the same distribution of availabilities. This distribution can also be denoted as *availability spectrum*. Availability spectrum c_i; $i = 0, 1, \ldots, n-1$ of a codeword $\boldsymbol{c}$ from [14, 4, 7] code is presented in Table 1. This availability spectrum contains strict availability for locality 2 and loose availabilities for localities 3, 4, …, 11.

From the theoretical point of view the symmetry of this availability spectrum is noticeable. For practical applications it is probably more important to mention that behind the strict availability 6 for locality 2, there are also repair sets with higher localities which can provide numerous loose availabilities.

Table 1. Availability versus locality distribution

Locality	Availability
2	6
3	22
4	40
5	72
6	116
7	116
8	72
9	40
10	22
11	6

Table 2. Locality two groups for each symbol in each codeword specified by indices in (4)-(6)

Index of symbol in codeword	**Locality two groups given by pairs of symbol indexes**
0	{2, 11}, {5, 6}, {3, 8}, {9, 10}, {4, 13}, {1, 7}
1	{2, 12}, {5, 13}, {4, 6}, {0, 7}, {8, 10}, {3, 9}
2	{10, 13}, {0, 11}, {1, 12}, {3, 6}, {5, 8}, {4, 9}
3	{4, 12}, {5, 11}, {7, 10}, {0, 8}, {1, 9}, {2, 6}
4	{10, 11}, {0, 13}, {3, 12}, {1, 6}, {5, 7}, {2, 9}
5	{10, 12}, {1, 13}, {3, 11}, {0, 6}, {4, 7}, {2, 8}
6	{1, 4}, {2, 3}, {0, 5}, {7, 13}, {9, 12}, {8, 11}
7	{11, 12}, {0, 1}, {3, 10}, {4, 5}, {6, 13}, {8, 9}
8	{12, 13}, {1, 10}, {0, 3}, {2, 5}, {6, 11}, {7, 9}
9	{11, 13}, {0, 10}, {1, 3}, {2, 4}, {6, 12}, {7, 8}
10	{2, 13}, {4, 11}, {5, 12}, {3, 7}, {1, 8}, {0, 9}
11	{0, 2}, {4, 10}, {3, 5}, {7, 12}, {6, 8}, {9, 13}
12	{1, 2}, {3, 4}, {5, 10}, {7, 11}, {8, 13}, {6, 9}
13	{2, 10}, {0, 4}, {1, 5}, {6, 7}, {8, 12}, {9, 11}

Table 2 contains information on which symbols participate in locality 2 repair sets for each codeword symbol valid for each codeword from the [14, 4, 7] code. These repair sets are specified by indexes corresponding to (4)-(6).

Table 2 shows that the locality 2 has strict availability 6 because for each codeword symbol there are 6 disjoint repair sets.

4 Conclusions

In this paper, the [14, 4, 7] binary linear block code was interpreted as a graph composed of four [7, 3, 4] Simplex codes graphs drawn on the faces of a tetrahedron in such a way that the vertices and edges of the graphs on the tetrahedron edges were merged. The resulting graph is illustrated in Fig. 2. The availability spectrum was introduced as a tool which can provide deeper insight into the distribution of different availabilities for each codeword symbol.

The computerized analyses of the [14, 4, 7] code revealed that all symbols of this code have the same symmetric availability distribution given by Tab. 1. The smallest locality in it is two. Each codeword symbol of this code can be recovered by interaction with two other symbols in 6 disjoint sets. Therefore the [14, 4, 7] code is (2, 6)-LRC. It is achieving upper bounds in [9] on k and d_m for codes with all-symbols locality 2 and strict availability 6 and $n = 14$.

It was also mentioned that higher values of localities with relatively high loose availabilities are present in this code, which could be of interest for practical applications. Not only data recovery but also the so-called hot data access could be satisfied by exploiting the properties of this code. The code could be used as a basic building block for scalable codes with locality and availability similar to the codes proposed in [5] but using 3-D graphs for their construction.

Acknowledgement

This work was supported by the Slovak Research and Development Agency under Contract no. APVV-19-0436 and by European Cooperation in Science and Technology funding agency under COST Action 22168 6G-PHYSEC.

References

[1] A. G. Dimakis, P. B. Godfrey, Y. Wu, M. J. Wainwright and K. Ramchandran, "Network Coding for Distributed Storage Systems," in *IEEE Transactions on Information Theory*, vol. 56, no. 9, pp. 4539-4551, Sept. 2010, doi: 10.1109/TIT.2010.2054295.

[2] E. Tomes and N. Altiparmak, "A Comparative Study of HDD and SSD RAIDs' Impact on Server Energy Consumption," *2017 IEEE International Conference on Cluster Computing (CLUSTER)*, Honolulu, HI, USA, 2017, pp. 625-626, doi: 10.1109/CLUSTER.2017.103.

[3] J. Wan, J. Wang, Q. Yang and C. Xie, "S2-RAID: A new RAID architecture for fast data recovery," *2010 IEEE 26th Symposium on Mass Storage Systems and Technologies (MSST)*, Incline Village, NV, USA, 2010, pp. 1-9, doi: 10.1109/MSST.2010.5496980.

[4] M. F. Aktaş, S. Kadhe, E. Soljanin and A. Sprintson, "Download Time Analysis for Distributed Storage Codes With Locality and Availability," in *IEEE Transactions on Communications*, vol. 69, no. 6, pp. 3898-3910, June 2021, doi: 10.1109/TCOMM.2021.3067054.

[5] P. Farkaš, "Scalable Family of Codes with Locality and Availability for Information Repair in Distributed Storage Systems," *2019 4th International Conference on Smart and Sustainable Technologies (SpliTech)*, Split, Croatia, 2019, pp. 1-4, doi: 10.23919/SpliTech.2019.8783148.

[6] N. Silberstein and A. Zeh, "Optimal binary locally repairable codes via anticodes," *2015 IEEE International Symposium on Information Theory (ISIT)*, Hong Kong, China, 2015, pp. 1247-1251, doi: 10.1109/ISIT.2015.7282655.

[7] M. Blaum, "Extended Integrated Interleaved Codes Over Any Field with Applications to Locally Recoverable Codes," in *IEEE Transactions on Information Theory*, vol. 66, no. 2, pp. 936-956, Feb. 2020, doi: 10.1109/TIT.2019.2934134.

[8] M. Grassl, "Code Tables: Bounds on parameters of various types of codes,"[Online]. Available: http://www.codetables.de/

[9] S. Kruglik, K. Nazirkhanova and A. Frolov, "New Bounds and Generalizations of Locally Recoverable Codes with Availability," in *IEEE Transactions on Information Theory*, vol. 65, no. 7, pp. 4156-4166, July 2019, doi: 10.1109/TIT.2019.2897705.

[10] S. Kadhe and R. Calderbank, "Rate optimal binary linear locally repairable codes with small availability," *2017 IEEE International Symposium on Information Theory (ISIT)*, Aachen, Germany, 2017, pp. 166-170, doi: 10.1109/ISIT.2017.8006511.

Received 1 December 2023