Analysis of the upper bound of error correction time for the data protection with the use of the Reed-Solomon codes

P A Rahman¹ and V E Bolnokin²

¹Department of Automated Technological and information Systems, Ufa State Petroleum Technological University, 2, October Avenue, Sterlitamak, 453118, Russian Federation
²Mechanical Engineering Research Institute, named A.A. Blagonravov of the Russian Academy of Sciences, 4, Maliy Haritonievsky str., Moscow, 101000, Russian Federation

E-mail: pavelar@yandex.ru, vitalybolnokin@yandex.ru

Abstract. This scientific paper deals with the error-correcting coding for the data protection and analysis of the upper bound of error correction time on application of the Reed-Solomon codes. The data encoding and decoding algorithms for the Reed-Solomon codes are briefly overviewed. The obtained by the author analytical formula for estimation of the upper bound of error correction time for the given size of data frames and multiplicity of correctable errors is presented. Finally, results of the experimental research of frame decoding and error correction time are also discussed.

1. Introduction
In present days the data transmission and storage systems [1-4] are still vulnerable to physical damages and external noises, which may cause corruption of the user data. Therefore, to protect the critical information in the data transmission and storage systems the different coding technologies based on the error-correcting codes are used. The Reed-Solomon codes [5, 6] are one of them. However, the Reed-Solomon codes use special algorithms for coding and decoding of data frames on application of the Galois Fields arithmetic [7, 8]. The encoding algorithm is relatively simple and it does not require significant time to encode the data frame. The decoding algorithm works fast only if the data frame is intact and the decoder detects only that the frame has no errors. Otherwise, if the data frame is corrupted, then it takes long time to calculate error locators and values.

One of the most important parameter for data coding on application of the Reed-Solomon codes is the multiplicity of correctable errors, i.e. maximum number of errors (corrupted symbols), which can be properly detected and corrected by the decoding algorithm. On the one hand an increase of the multiplicity of correctable errors significantly improves the error-tolerance of data frames. On the other hand, it also increases the calculation time of the error locators and values, which leads to loss of performance in the data transmission and storage systems.

Within the research work in the field of reliability of the data transmission and storage systems [9, 10] the author raised a scientific task of analysis of the upper bound of error correction time on application of the Reed-Solomon codes for the given size of data frames and multiplicity of
correctable errors. The author obtained an analytical formula for estimation of the upper bound and carried out experimental research of the frame decoding and error correction time.

2. The encoding and decoding algorithms for the Reed-Solomon codes

In most cases of the redundant coding on application of the Reed-Solomon codes the user data are represented as blocks of $k$ bytes, and these blocks are interpreted as the polynomials $M(x)$ over the Galois Field $GF(2^8)$ and the bytes within the block are interpreted as the elements of the field:

$$M(x) = M_{k-1}x^{k-1} + \ldots + M_1x + M_0; \quad M_j \in GF(2^8).$$

Let us briefly overview the encoding algorithm [5, 6]. At first, encoder calculates the control polynomial $R(x)$ for the given multiplicity $t$ of correctable errors:

$$R(x) = (x'M(x)) \mod g(x).$$

Here, $r = 2t$ is number of the control bytes and $g(x)$ is the generative polynomial for the Reed-Solomon codes, which is calculated as:

$$g(x) = \prod_{s=1}^{r} (x + \alpha^s).$$

Here, $\alpha$ is the primitive element of the Galois field $GF(2^8)$. Finally, the encoder calculates the data frame polynomial $F(x)$:

$$F(x) = x'M(x) + R(x).$$

The coefficients of the obtained polynomial are interpreted as the bytes of the resultant data frame. The size of data frames is equal to the $n = k + r$. The maximum size of data frames is $2^8 - 1 = 255$ bytes for the Reed-Solomon codes over the Galois field $GF(2^8)$.

Let us briefly overview the decoding algorithm [5, 6]. The decoding algorithm deals with some data frame, which mathematically can be interpreted as the sum $C(x)$ of the original data frame polynomial $F(x)$ and unknown error polynomial $E(x)$:

$$C(x) = F(x) + E(x).$$

At first, decoder calculates the syndrome components $S_j$:

$$S_j = C(\alpha^j); \quad j = 1 \ldots r.$$  

If all of the syndrome components are zero, $S_j = 0$, then the decoder makes decision that the data frame has no errors. Otherwise, the decoder tries to find the error locator polynomial $\Lambda(x)$ with the minimal degree $\tau$, which fulfills the following system of linear algebraic equations:

$$S_j = \sum_{i=1}^{\tau} \Lambda_i S_{j-i}; \quad j = \tau + 1 \ldots r.$$  

It should be noted, that $\tau$ is interpreted by the decoder as the assumed number of errors.

To find the error locator polynomial $\Lambda(x)$ with the minimal degree $\tau$ the well-known Berlekamp-Massey algorithm [5, 6] is used. If the error locator polynomial $\Lambda(x)$ cannot be found, then the decoder makes decision that the data frame is uncorrectable. Otherwise, the decoder tries to find the roots $x^*$ of the error locator polynomial by solving the equation $\Lambda(x) = 0$.

If the roots $x^*$ are found and their quantity is not equal to $\tau$, then the decoder makes decision that the data frame is uncorrectable. Otherwise, the decoder calculates the error locators:
\[ u_i = \log_2(1 / x_i^*) \]; \( l = 1 \ldots \tau \). \hfill (7)

If one of locators is outside the data frame, \( u_i \notin [0 \ldots n - 1] \), then the decoder makes decision that the data frame is uncorrectable. Otherwise, the magnitude polynomial \( \Omega(x) \) and formal derivative of the error locator polynomial \( \Lambda(x) \) are calculated by using the following formulas:

\[ \Omega(x) = \sum_{q=0}^{t-1} x^q \sum_{i=0}^{q} \Lambda_i S_{q-i+1}; \hfill (8) \]

\[ \Lambda'(x) = \sum_{i=1}^{\tau} \Lambda_i(i \mod 2)x^{i-1}. \]

Next, the decoder calculates the error values by using the obtained polynomials and roots \( x^* \):

\[ v_j = \Omega(x_j^*) / \Lambda'(x_j^*); \hfill (9) \]

Finally, the decoder forms the error polynomial and corrects the data frame polynomial:

\[ \tilde{F}(x) = C(x) + \tilde{E}(x); \quad \tilde{E}(x) = \sum_{j=0}^{\tau} v_j x^{u_j}. \hfill (10) \]

### 3. Analytical estimation of the upper bound of error correction time

Study of the decoding algorithm shows that if the data frame has no errors, then the decoder only need to calculate the syndrome components. Therefore, this is the fastest case of the data frame decoding.

Next, if the actual number of errors is greater than the multiplicity of correctable errors, in several cases it is possible that the decoder will fail to resolve mathematically the decoding task and provide the error correction, and the decoding task will be incomplete.

At last, if the actual number of errors is less or equal to the multiplicity of correctable errors, in this case the decoder always will provide all of the decoding steps, including the error correction. Moreover, if the actual number of errors is exactly equal to the multiplicity of correctable errors, obviously, in this case the maximum number of calculations will be provided, so this case is the «worst» from the viewpoint of the computational complexity of the frame decoding algorithm.

On basis of the advanced analysis of the decoding algorithm for the worst case, when the actual number of errors in data frames is exactly equal to the multiplicity of correctable errors, the author obtained the formula for the upper bound of computational complexity of the error correction:

\[ C_{\text{max}} = \frac{71}{2} t^2 + \left(10n + \frac{2647}{2}\right) t + 2n + 2295. \hfill (11) \]

Here, \( n \) is size of data frames and \( t \) is multiplicity of correctable errors.

It should be noted, that the obtained formula actually estimates the total number of the arithmetic operations in the Galois Field \( \text{GF}(2^k) \), which are required for the frame decoding and error correction, and considers that the multiplication and division operations on average have four time more complexity relative to the addition operation.

Next, to obtain the estimation of the upper bound of error correction time (in microseconds) we need to divide the upper bound of computational complexity by some performance coefficient \( W \):

\[ T_{\text{max}} = C_{\text{max}} / W. \hfill (12) \]

The performance coefficient is a specific parameter, which depends on the architecture and performance of the computing system, on which the Reed-Solomon coding and decoding software is running. In particular, for a computer based on the Intel® Pentium™ IV processor with the clock frequency 3.0 GHz the author obtained the following value of the performance coefficient:
Table 1 shows the results of calculation of the upper bound of error correction time for several sizes of data frames and multiplicities of correctable errors by the formula 11 and 12.

Table 1. Results of calculation of the upper bound of error correction time.

| n   | Upper bound of error correction time, us |
|-----|----------------------------------------|
| 32  | 14.473 20.746 34.053 63.724 |
| 64  | 15.849 23.269 38.871 73.129 |
| 96  | 17.226 25.792 43.688 82.534 |
| 128 | 18.602 28.315 48.505 91.939 |
| 160 | 19.978 30.839 53.326 101.344 |
| 192 | 21.355 33.362 58.139 110.749 |
| 224 | 22.731 35.885 62.957 120.154 |
| 255 | 24.064 38.329 67.624 129.265 |

4. Experimental research of the frame decoding and error correction time

To provide the experimental research of the frame decoding and error correction time the author developed special software, which provides generation, encoding, corruption and decoding of data frames on application of the Reed-Solomon codes.

Figure 1 shows the main window of the developed software. The developed software allows to generate and encode the given number of data frames with the size of $1 \leq n \leq 255$ bytes and multiplicity of correctable errors $1 \leq t \leq 127$, corrupt the given fixed number of bytes $w$ in random positions of data frames and using the random error values, decode data frames and collect information about the average encoding and decoding time.

![Figure 1. Main window of the developed software for the experimental research.](image-url)
To provide the experimental research of the frame decoding time and find out the worst case, when the decoding time is the highest, the author carried out several series of experiments for the fixed frame size $n = 255$, several multiplicities of correctable errors $t = 1, 2, 4$ and $8$, and different number of corrupted bytes $w = 1 \ldots 12$. In each series 1000000 frames were generated, encoded, corrupted, decoded and the average frame decoding time was measured by using the developed software on the computer based on the Intel® Pentium™ IV 3.0 GHz processor.

The results of the experimental research of the frame decoding time are shown in table 2 and they are also graphically presented in figure 2. It is easy to see, that for the given frame size $n$ and multiplicity of correctable errors $t$ the frame decoding time reaches the highest value when the number of corrupted bytes $w$ is equal to the multiplicity of correctable errors.

**Table 2.** Average frame decoding time for the fixed size $n = 255$ of data frames, several multiplicities of correctable errors $t$ and different number of corrupted bytes $w$.

| $w$ | $t = 1$  | $t = 2$  | $t = 4$  | $t = 8$  |
|-----|----------|----------|----------|----------|
| 0   | 6.394    | 10.571   | 19.021   | 36.729   |
| 1   | **24.259** | 33.372   | 51.661   | 90.292   |
| 2   | 24.136   | **36.847** | 55.462   | 94.629   |
| 3   | 24.113   | 30.524   | 60.409   | 99.781   |
| 4   | 24.039   | 30.529   | **66.391** | 105.784 |
| 5   | 24.058   | 30.574   | 45.464   | 111.613  |
| 6   | 24.011   | 30.582   | 45.464   | 116.344  |
| 7   | 24.005   | 30.508   | 45.606   | 120.811  |
| 8   | 24.063   | 30.435   | 45.602   | **126.426** |
| 9   | 24.013   | 30.569   | 45.593   | 83.857   |
| 10  | 24.055   | 30.434   | 45.553   | 83.888   |
| 11  | 24.036   | 30.501   | 45.514   | 83.845   |
| 12  | 24.061   | 30.632   | 45.546   | 83.848   |

**Figure 2.** Graphs of the average frame decoding time for the fixed size $n = 255$ of data frames, several multiplicities of correctable errors $t$ and different number of corrupted bytes $w$. 


To provide the experimental research of error correction time in the worst case, when the actual number of errors \( w \) is equal to the multiplicity \( t \) of correctable errors, the author also carried out several series of experiments for the different frame sizes \( n \) and several multiplicities of the correctable errors \( t = 1, 2, 4 \) and 8. In each series 1000000 frames were generated, encoded, corrupted, decoded and the average frame decoding time was measured.

The results of experimental research of the average frame decoding time in the worst case \( (w = t) \) are shown in Table 3. It is easily seen that the measured values are in good accordance with the values in table 1, obtained by the formula for the upper bound of error correction time.

| \( n \) | Average frame decoding time, us |
|---|---|
| \( t = 1 \) | \( t = 2 \) | \( t = 4 \) | \( t = 8 \) |
| 32 | 14.078 | 19.488 | 34.293 | 62.087 |
| 64 | 15.621 | 21.971 | 39.233 | 71.266 |
| 96 | 17.056 | 24.465 | 43.602 | 80.532 |
| 128 | 18.535 | 26.986 | 48.014 | 89.545 |
| 160 | 19.936 | 29.562 | 52.803 | 98.571 |
| 192 | 21.444 | 31.965 | 57.355 | 108.054 |
| 224 | 22.829 | 34.414 | 62.003 | 116.666 |
| 255 | 24.259 | 36.847 | 66.391 | 126.426 |

5. Conclusion

Thus, within the scope of this scientific research the author analyzed the upper bound of the error correction time on application of the Reed-Solomon codes for the given size of data frames and multiplicity of correctable errors and obtained analytical formula for estimation of the upper bound and carried out experimental research of the frame decoding and error correction time.

The obtained results were used for development of the specialized software and laboratory works aimed to analyze the data frame encoding and decoding time on application of the Reed-Solomon codes for the students of technical specialties.

Acknowledgements

The author is grateful to professor I. I. Ladygin from Moscow Power Engineering Institute for scientific support and theoretical base in the field of error-correcting codes.

References

[1] Tanenbaum A S and Wetherall D J 2010 Computer Networks (New Jersey: Prentice Hall PTR)
[2] Nedyalkov I, Stefanov A and Georgiev G 2019 International Journal of Information Technologies and Security 11(1) 3-12
[3] Nelson S 2011 Pro data backup and recovery (Apress)
[4] Thomasian A and Blaum M 2009 ACM Transactions on Storage 5(3) 7
[5] Moon T K 2005 Error correcting coding: mathematical methods and algorithms (New Jersey: John Wiley & Sons Inc.)
[6] Wicker S B and Bhargava V K 1999 Reed-Solomon codes and their application (Wiley-IEEE Press)
[7] Mullen G L and Panario D 2013 Handbook of Finite Fields (CRC Press)
[8] Lidl R and Niederreiter H 1983 Finite fields (Reading, MA: Addison-Wesley)
[9] Rahman P A 2018 J. Phys.: Conf. Ser. 1015 022016
[10] Rahman P A 2017 IOP Conf. Ser.: Mater. Sci. Eng. 177 012088